Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods

Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods

Nicolas Loizou * Peter Richtárik†

December 22, 2017‡

Abstract

In this paper we study several classes of stochastic optimization algorithms enriched with *heavy ball momentum*. Among the methods studied are: stochastic gradient descent, stochastic Newton, stochastic proximal point and stochastic dual subspace ascent. This is the first time momentum variants of several of these methods are studied. We choose to perform our analysis in a setting in which all of the above methods are equivalent. We prove global nonassymptotic linear convergence rates for all methods and various measures of success, including primal function values, primal iterates (in L2 sense), and dual function values. We also show that the primal iterates converge at an accelerated linear rate in the L1 sense. This is the first time a linear rate is shown for the stochastic heavy ball method (i.e., stochastic gradient descent method with momentum). Under somewhat weaker conditions, we establish a sublinear convergence rate for Cesaro averages of primal iterates. Moreover, we propose a novel concept, which we call *stochastic momentum*, aimed at decreasing the cost of performing the momentum step. We prove linear convergence of several stochastic methods with stochastic momentum, and show that in some sparse data regimes and for sufficiently small momentum parameters, these methods enjoy better overall complexity than methods with deterministic momentum. Finally, we perform extensive numerical testing on artificial and real datasets, including data coming from average consensus problems.

Keywords stochastic methods · heavy ball momentum · linear systems · randomized coordinate descent · randomized Kaczmarz · stochastic gradient descent · stochastic Newton · quadratic optimization · convex optimization

Mathematical Subject Classifications 68Q25 · 68W20 · 68W40 · 65Y20 · 90C15 · 90C20 · 90C25 · 15A06 · 15B52 · 65F10

1 Introduction

Two of the most popular algorithmic ideas for solving optimization problems involving big volumes of data are *stochastic approximation* and *momentum*. By stochastic approximation we refer to the practice pioneered by Robins and Monro [65] of replacement of costly-to-compute quantities (e.g., gradient of the objective function) by cheaply-to-compute *stochastic* approximations thereof (e.g., unbiased estimate of the gradient). By momentum we refer to the *heavy ball*

*School of Mathematics, The University of Edinburgh. — E-mail: n.loizou@sms.ed.ac.uk

†CEMSE, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia — School of Mathematics, The University of Edinburgh, United Kingdom — Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, Moscow, Russia.— E-mail: peter.richtarik@ed.ac.uk

‡A short version of this paper (5 pages) was posted on arXiv on 30 Oct 2017 [39]. The paper was accepted for presentation at the 2017 NIPS Optimization for Machine Learning workshop in a peer reviewed process. The accepted papers are listed on the website of the workshop, but are not published in any proceedings volume.

technique originally developed by Polyak [54] to accelerate the convergence rate of gradient-type methods.

While much is known about the effects of *stochastic approximation* and *momentum* in isolation, surprisingly little is known about the *combined effect* of these two popular algorithmic techniques. For instance, to the best of our knowledge, there is no context in which a method combining stochastic approximation with momentum is known to have a linear convergence rate. One of the contributions of this work is to show that there are important problem classes for which a linear rate can indeed be established for a range of stepsize and momentum parameters.

1.1 Setting

In this paper we study three closely related problems:

(i) stochastic optimization,

(ii) best approximation, and

(iii) (bounded) concave quadratic maximization.

These problems and the relationships between them are described in detail in Section 3. Here we only briefly outline some of the key relationships. By stochastic optimization we refer to the problem of the form

$$\min_{x\in\mathbb{R}^n} f(x) := \mathbb{E}\left[f_{\mathbf{S}}(x)\right], \tag{1}$$

where the expectation is over random matrices $\mathbf{S}$ drawn from an arbitrary distribution $\mathcal{D}$, and $f_{\mathbf{S}}$ is a stochastic convex quadratic function of a least-squares type, depending on $\mathbf{S}$, and in addition on a matrix $\mathbf{A}\in\mathbb{R}^{m\times n}$, vector $b\in\mathbb{R}^m$, and an $n\times n$ positive definite matrix $\mathbf{B}$ (see Section 3 for full details). The problem is constructed in such a way that the set of minimizers of f is identical to the set of solutions of a given (consistent) linear system

$$\mathbf{A}x = b, \tag{2}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}$ and $b\in\mathbb{R}^m$. In this sense, (1) can be seen as the reformulation of the linear system (2) into a stochastic optimization problem. Such reformulations provide an explicit connection between the fields of linear algebra and stochastic optimization, which may inspire future research by enabling the transfer of knowledge, techniques, and algorithms from one field to another. For instance, the randomized Kaczmarz method of Strohmer and Vershynin [69] for solving (2) is equivalent to the stochastic gradient descent method applied to (1), with $\mathcal{D}$ corresponding to a discrete distribution over unit coordinate vectors in $\mathbb{R}^m$. However, the flexibility of being able to choose $\mathcal{D}$ arbitrarily allows for numerous generalizations of the randomized Kaczmarz method [64]. Likewise, provably faster variants of the randomized Kaczmarz method (for instance, by utilizing importance sampling) can be designed using the connection.

1.2 Three stochastic methods

Problem (1) has several peculiar characteristics which are of key importance to this paper. For instance, the Hessian of $f_{\mathbf{S}}$ is a (random) projection matrix, which can be used to show that $f_{\mathbf{S}}(x) = \frac{1}{2}\|\nabla f_{\mathbf{S}}(x)\|^2_{\mathbf{B}}$. Moreover, it follows that the Hessian of f has all eigenvalues bounded by 1, and so on. These characteristics can be used to show that several otherwise *distinct* stochastic algorithms for solving the stochastic optimization problem (1) are *identical* [64]. In particular, the following optimization methods for solving (1) are identical[1]:

[1] In addition, these three methods are identical to a stochastic fixed point method (with relaxation) for solving the fixed point problem $x = \mathbb{E}\left[\Pi_{\mathcal{L}_{\mathbf{S}}}(x)\right]$, where $\mathcal{L}_{\mathbf{S}}$ is the set of solutions of $\mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b$, which is a *sketched* version of the linear system (2), and can be seen as a stochastic approximation of the set $\mathcal{L} := \{x \,:\, \mathbf{A}x = b\}$

- *Method 1:* stochastic gradient descent (SGD),
- *Method 2:* stochastic Newton method (SN), and
- *Method 3:* stochastic proximal point method (SPP);

all with a fixed stepsize $\omega > 0$. The methods will be described in detail in Section 3; see also Table 2 for a quick summary.

The equivalence of these methods is useful for the purposes of this paper as it allows us to study their variants *with momentum* by studying a single algorithm only. We are not aware of any successful attempts to analyze momentum variants of SN and SPP and as we said before, there are no linearly convergent variants of SGD with momentum in *any setting*.

1.3 Best approximation, duality and stochastic dual subspace ascent

It was shown in [24] in the $\omega = 1$ case and in [64] in the general $\omega > 0$ case that SGD, SN and SPP converge to a very particular minimizer of f: the projection of the starting point x_0 onto the solution set of the linear system (2). This naturally leads to the best approximation problem, which is the problem of projecting[2] a given vector onto the solution space of the linear system (2):

$$\min_{x\in\mathbb{R}^n} \tfrac{1}{2}\|x - x_0\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{A}x = b. \tag{3}$$

The dual of the best approximation problem is an unconstrained concave quadratic maximization problem [24]. Consistency of $\mathbf{A}x = b$ implies that the dual is bounded. It follows from the results of [24] that for $\omega = 1$, the random iterates of SGD, SN and SPP arise as affine images of the random iterates produced by an algorithm for solving the dual of the best approximation problem (3), known as

- *Method 4:* stochastic dual subspace ascent (SDSA).

In this paper we show that this equivalence extends beyond the $\omega = 1$ case, specifically for $0 < \omega < 2$, and further study *SDSA with momentum*. We then show that SGD, SN and SPP with momentum arise as affine images of SDSA with momentum. SDSA proceeds by taking steps in a random subspace spanned by the columns of $\mathbf{S}$ randomly drawn in each iteration from $\mathcal{D}$. In this subspace, the method moves to the point which maximizes the dual objective, $D(y)$. Since $\mathcal{D}$ is an arbitrary distribution of random matrices, SDSA moves in arbitrary random subspaces, and as such, can be seen as a vast generalization of randomized coordinate descent methods and their minibatch variants [16, 58].

1.4 Structure of the paper

The remainder of this work is organized as follows. In Section 2 we summarize our contributions in the context of existing literature. In Section 3 we provide a detailed account of the stochastic optimization problem, the best approximation and its dual. Here we also describe the SGD, SN and SPP methods. In Section 4 we describe and analyze primal methods with momentum (mSGD, mSN and mSPP), and in Section 5 we describe and analyze the dual method with momentum (mSDSA). In Section 6 we describe and analyze primal methods with stochastic momentum (smSGD, smSN and smSPP). Numerical experiments are presented in Section 8. Proofs of all key results can be found in the appendix.

[2] In the rest of the paper we consider projection with respect to an arbitrary Euclidean norm.

1.5 Notation

The following notational conventions are used in this paper. Boldface upper-case letters denote matrices; $\mathbf{I}$ is the identity matrix. By $\mathcal{L}$ we denote the solution set of the linear system $\mathbf{A}x = b$. By $\mathcal{L}_{\mathbf{S}}$, where $\mathbf{S}$ is a random matrix, we denote the solution set of the *sketched* linear system $\mathbf{S}^\top \mathbf{A}x = \mathbf{S}^\top b$. Throughout the paper, $\mathbf{B}$ is an $n \times n$ positive definite matrix giving rise to an inner product and norm on $\mathbb{R}^n$. Unless stated otherwise, throughout the paper, x_* is the projection of x_0 onto $\mathcal{L}$ in the $\mathbf{B}$-norm: $x_* = \Pi^{\mathbf{B}}_{\mathcal{L}}(x_0)$. We write $[n] := \{1, 2, \ldots, n\}$.

2 Momentum Methods and Our Contributions

In this section we give a brief review of the relevant literature, and provide a summary of our contributions.

2.1 Heavy ball method

The baseline first-order method for minimizing a differentiable function f is the gradient descent (GD) method,

$$x_{k+1} = x_k - \omega_k \nabla f(x_k),$$

where $\omega_k > 0$ is a stepsize. For convex functions with L-Lipschitz gradient (function class $\mathcal{F}^{1,1}_{0,L}$), GD converges at at the rate of $\mathcal{O}(L/\epsilon)$. When, in addition, f is μ-strongly convex (function class $\mathcal{F}^{1,1}_{\mu,L}$), the rate is linear: $\mathcal{O}((L/\mu)\log(1/\epsilon))$ [48]. To improve the convergence behavior of the method, Polyak proposed to modify GD by the introduction of a (heavy ball) momentum term[3], $\beta(x_k - x_{k-1})$. This leads to the gradient descent method with momentum (mGD), popularly known as the *heavy ball method:*

$$x_{k+1} = x_k - \omega_k \nabla f(x_k) + \beta(x_k - x_{k-1}).$$

More specifically, Polyak proved that with the correct choice of the stepsize parameters ω_k and momentum parameter β, a *local* accelerated linear convergence rate of $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$ can be achieved in the case of twice continuously differentiable, μ-strongly convex objective functions with L-Lipschitz gradient (function class $\mathcal{F}^{2,1}_{\mu,L}$). See the first line of Table 1.

Recently, Ghadimi et al. [20] performed a *global* convergence analysis for the heavy ball method. In particular, the authors showed that for a certain combination of the stepsize and momentum parameter, the method converges sublinearly to the optimum when the objective function is convex and has Lipschitz gradient ($f \in \mathcal{F}^{1,1}_{0,L}$), and linearly when the function is also strongly convex ($f \in \mathcal{F}^{1,1}_{\mu,L}$). A particular, selection of the parameters ω and β that gives the desired accelerated linear rate was not provided.

To the best of our knowledge, despite considerable amount of work on the on heavy ball method, there is still no global convergence analysis which would guarantee an accelerated linear rate for $f \in \mathcal{F}^{1,1}_{\mu,L}$. However, in the special case of a strongly convex quadratic, an elegant proof was recently proposed in [35]. Using the notion of integral quadratic constraints from robust control theory, the authors proved that by choosing $\omega_k = \omega = 4/(\sqrt{L} + \sqrt{\mu})^2$ and

[3] Arguably a much more popular, certainly theoretically much better understood alternative to Polyak's momentum is the momentum introduced by Nesterov [46, 48], leading to the famous *accelerated gradient descent (AGD)* method. This method converges nonassymptotically and globally; with optimal sublinear rate $\mathcal{O}(\sqrt{L/\epsilon})$ [45] when applied to minimizing a smooth convex objective function (class $\mathcal{F}^{1,1}_{0,L}$), and with the optimal linear rate $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$ when minimizing smooth strongly convex functions (class $\mathcal{F}^{1,1}_{\mu,L}$). Both Nesterov's and Polyak's update rules are known in the literature as "momentum" methods. In this paper, however, we focus exclusively on Polyak's heavy ball momentum.

$\beta = (\sqrt{L/\mu}-1)^2/(\sqrt{L/\mu}+1)^2$, the heavy ball method enjoys a global *asymptotic* accelerated convergence rate of $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$. The aforementioned results are summarized in the first part of Table 1.

Extensions of the heavy ball method have been recently proposed in the proximal setting [51], non-convex setting [52, 78] and for distributed optimization [21].

2.2 Stochastic heavy ball method

In contrast to the recent advances in our theoretical understanding of the (classical) heavy ball method, there has been less progress in understanding the convergence behavior of *stochastic* variants of the heavy ball method. The key method in this category is stochastic gradient descent with momentum (mSGD; aka: stochastic heavy ball method):

$$x_{k+1} = x_k - \omega_k g(x_k) + \beta(x_k - x_{k-1}),$$

where g_k is an unbiased estimator of the true gradient $\nabla f(x_k)$. While mSGD is used extensively in practice, especially in deep learning [70, 71, 33, 74], its convergence behavior is not very well understood.

In fact, we are aware of only two papers, both recent, which set out to study the complexity of mSGD: the work of Yang et al. [77], and the work of Gadat et al. [18]. In the former paper, a unified convergence analysis for stochastic gradient methods with momentum (heavy ball and Nesterov's momentum) was proposed; and an analysis for both convex and non convex functions was performed. For a general Lipschitz continuous convex objective function with bounded variance, a rate of $O(1/\sqrt{\epsilon})$ was proved. For this, the authors employed a decreasing stepsize strategy: $\omega_k = \omega_0/\sqrt{k+1}$, where ω_0 is a positive constant. In [18], the authors first describe several almost sure convergence results in the case of general non-convex coercive functions, and then provided a complexity analysis for the case of quadratic strongly convex function. However, the established rate is slow. More precisely, for strongly convex quadratic and coercive functions, mSGD with diminishing stepsizes $\omega_k = \omega_0/k^\beta$ was shown to convergence as $\mathcal{O}(1/k^\beta)$ when the momentum parameter is $\beta < 1$, and with the rate $O(1/\log k)$ when $\beta = 1$. The convergence rates established in both of these papers are sublinear. In particular, no insight is provided into whether the inclusion of the momentum term provides what is was aimed to provide: acceleration.

The above results are summarized in the second part of Table 1. From this perspective, our contribution lies in providing an in-depth analysis of mSGD (and, additionally, of SGD with stochastic momentum). Our contributions are discussed next.

2.3 Connection to incremental gradient methods

Assuming $\mathcal{D}$ is discrete distribution (i.e., we sample from M matrices, $\mathbf{S}^1, \dots, \mathbf{S}^M$, where $\mathbf{S}^i$ is chosen with probability $p_i > 0$), we can write the stochastic optimization problem (1) in the *finite-sum* form

$$\min_{x\in\mathbb{R}^n} f(x) = \sum_{i=1}^{M} p_i f_{\mathbf{S}^i}(x). \tag{4}$$

Choosing $x_0 = x_1$, mSGD with fixed stepsize $\omega_k = \omega$ applied to (4) can be written in the form

$$x_{k+1} = x_k - \omega\sum_{t=1}^{k}\beta^{k-t}\nabla f_{\mathbf{S}_t}(x_t) + \beta^k(x_1 - x_0) = x_k - \omega\sum_{t=1}^{k}\beta^{k-t}\nabla f_{\mathbf{S}_t}(x_t), \tag{5}$$

Method	Paper	Rate	Assumptions on f	Convergence
Heavy Ball (mGD)	Polyak, 1964 [54]	accelerated linear	$\mathcal{F}^{2,1}_{\mu,L}$	local
	Ghadimi et al, 2014 [20]	sublinear	$\mathcal{F}^{1,1}_{L}$	global
	Ghadimi et al, 2014 [20]	linear	$\mathcal{F}^{1,1}_{\mu,L}$	global
	Lessard et al, 2016 [35]	accelerated linear	$\mathcal{F}^{1,1}_{\mu,L}$ + quadratic	global, asymptotic
Stochastic Heavy Ball (mSGD)	Yang et al. 2016 [77]	sublinear	$\mathcal{F}^{1,1}_{0,L}$ + bounded variance	global, non-asymptotic
	Gadat et al, 2016 [18]	sublinear	$\mathcal{F}^{1,1}_{\mu,L}$ + other assumptions	global, non-asymptotic
	THIS PAPER	**see Table 3**	$\mathcal{F}^{1,1}_{0,L}$ + quadratic	global, non-asymptotic

Table 1: Known complexity results for gradient descent with momentum (mGD, aka: heavy ball method), and stochastic gradient descent with momentum (mSGD, aka: stochastic heavy ball method). We give the first linear and accelerated rates for mSGD. For full details on iteration complexity results we obtain, refer to Table 3.

where $\mathbf{S}_t = \mathbf{S}^i$ with probability p_i. Problem (4) can be also solved using incremental average/aggregate gradient methods, such as the IAG method of Blatt et al. [3]. These methods have a similar form to (5), with the main difference being in the way the past gradients are aggregated. While (5) uses a geometric weighting of the gradients, the incremental average gradient methods use a uniform/arithmetic weighting. The stochastic average gradient (SAG) method of Schmidt et al. [66] can be also written in a similar form. Note that mSGD uses a geometric weighting of previous gradients, while the the incremental and stochastic average gradient methods use an arithmetic weighting. Incremental and incremental average gradient methods are widely studied algorithms for minimizing objective functions which can expressed as a sum of finite convex functions. For a review of key works on incremental methods and a detailed presentation of the connections with stochastic gradient descent, we refer the interested reader to the excellent survey of Bertsekas [2]; see also the work of Tseng [72].

In [27], an incremental average gradient method with momentum was proposed for minimizing strongly convex functions. It was proved that the method converges to the optimum with linear rate. The rate is always worse than that of the no-momentum variant. However, it was shown experimentally that in practice the method is faster, especially in problems with high condition number. In our setting, the objective function has a very specifc structure (1). It is not a finite sum problem as the distribution $\mathcal{D}$ could be continous; and we also do not assume strong convexity. Thus, the convergence analysis of [27] can not be directly applied to our problem.

2.4 Summary of contributions

We now summarize the contributions of this paper.

New momentum methods. We study several classes of stochastic optimization algorithms (SGD, SN, SPP and SDSA) *with momentum*, which we call mSGD, mSN, mSPP and mSDSA, respectively (see the first and second columns of Table 2). We do this in a simplified setting with quadratic objectives where all of these algorithms are equivalent. These methods can be seen as solving three related optimization problems: the stochastic optimization problem (1), the best approximation problem (3) and its dual. To the best of our knowledge, momentum variants of SN, SPP and SDSA were not analyzed before.

Linear rate. We prove several (global and non-asymptotic) linear convergence results for our primal momentum methods mSGD/mSN/mSPP. First, we establish a linear rate for the decay of $\mathbb{E}\left[\|x_k - x_*\|^2_{\mathbf{B}}\right]$ to zero (i.e., $L2$ convergence), for a range of stepsizes $\omega > 0$ and momentum parameters $\beta \geq 0$. We show that the same rate holds for the decay of the expected

no momentum ($\beta = 0$)	momentum ($\beta \geq 0$)	stochastic momentum ($\beta \geq 0$)
SGD [23, $\omega = 1$], [64, $\omega > 0$] $x_{k+1} = x_k - \omega \nabla f_{\mathbf{S}_k}(x_k)$	**mSGD** [Sec 4] $+\beta(x_k - x_{k-1})$	**smSGD** [Sec 6] $+n\beta e_{i_k}^\top (x_k - x_{k-1}) e_{i_k}$
SN [64] $x_{k+1} = x_k - \omega(\nabla^2 f_{\mathbf{S}_k}(x_k))^{\dagger_{\mathbf{B}}} \nabla f_{\mathbf{S}_k}(x_k)$	**mSN** [Sec 4] $+\beta(x_k - x_{k-1})$	**smSN** [Sec 6] $+n\beta e_{i_k}^\top (x_k - x_{k-1}) e_{i_k}$
SPP [64] $x_{k+1} = \arg\min_x \left\{ f_{\mathbf{S}_k}(x) + \frac{1-\omega}{2\omega} \|x - x_k\|^2_{\mathbf{B}} \right\}$	**mSPP** [Sec 4] $+\beta(x_k - x_{k-1})$	**smSPP** [Sec 6] $+n\beta e_{i_k}^\top (x_k - x_{k-1}) e_{i_k}$
SDSA [24, $\omega = 1$] $y_{k+1} = y_k + \mathbf{S}_k \lambda_k$	**mSDSA** [Sec 5] $+\beta(y_k - y_{k-1})$	

Table 2: All methods analyzed in this paper. The methods highlighted in bold (with momentum and stochastic momentum) are new. SGD = Stochastic Gradient Descent, SN = Stochastic Newton, SPP = Stochastic Proximal Point, SDSA = Stochastic Dual Subspace Ascent. At iteration k, matrix $\mathbf{S}_k$ is drawn in an i.i.d. fashion from distribution $\mathcal{D}$, and a stochastic step is performed.

function values $\mathbb{E}[f(x_k) - f(x_*)]$ of (1) to zero. Further, the same rate holds for mSDSA, in particular, this is for the convergence of the dual objective to the optimum. For a summary of these results, and pointers to the relevat theorems, refer to lines 1, 2 and 6 of Table 3. Unfortunately, the theoretical rate for all our momentum methods is optimized for $\beta = 0$, and gets worse as the momentum parameter increases. However, no prior linear rate for any of these methods with momentum are known. We give the first linear convergence rate for SGD with momentum (i.e., for the stochastic heavy ball method).

Accelerated linear rate. We then study the decay of the larger quantity $\|\mathbb{E}[x_k] - x_*\|^2_{\mathbf{B}}$ to zero (i.e., L1 convergence). In this case, we establish an *accelerated* linear rate, which depends on the square root of the condition number (of the Hessian of f). This is a quadratic speedup when compared to the no-momentum methods as these depend on the condition number. See lines 4 and 5 of Table 3. To the best of our knowledge, this is the first time an accelerated rate is obtained for the stochastic heavy ball method (mSGD). Note that there are no global non-asymptotic accelerated linear rates proved even in the non-stochastic setting (i.e., for the heavy ball method). Moreover, we are not aware of any accelerated linear convergence results for the stochastic proximal point method.

Sublinear rate for Cesaro averages. We show that the Cesaro averages, $\hat{x}_k = \frac{1}{k} \sum_{t=0}^{k-1} x_t$, of all primal momentum methods enjoy a sublinear $O(1/k)$ rate (see line 3 of Table 3). This holds under weaker assumptions than those which lead to the linear convergence rate.

Primal-dual correspondence. We show that SGD, SN and SPP with momentum arise as affine images of SDSA with momentum (see Theorem 5). This extends the result of [24]

Algorithm	ω	momentum β	Quantity converging to 0	Rate (all: global, non-asymptotic)	Theorem
mSGD/mSN/mSPP	$(0,2)$	≥ 0	$\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}]$	linear	1
mSGD/mSN/mSPP	$(0,2)$	≥ 0	$\mathbb{E}[f(x_k) - f(x_*)]$	linear	1
mSGD/mSN/mSPP	$(0,2)$	≥ 0	$\mathbb{E}\left[f(\hat{x}_k)\right] - f(x_*)$	sublinear: $O(1/k)$	3
mSGD/mSN/mSPP	1	$\left(1 - \sqrt{0.99\lambda^+_{\min}}\right)^2$	$\|\mathbb{E}[x_k - x_*]\|^2_{\mathbf{B}}$	accelerated linear	4
mSGD/mSN/mSPP	$\frac{1}{\lambda_{\max}}$	$\left(1 - \sqrt{0.99\frac{\lambda^+_{\min}}{\lambda_{\max}}}\right)^2$	$\|\mathbb{E}[x_k - x_*]\|^2_{\mathbf{B}}$	accelerated linear (better than for $\omega = 1$)	4
mSDSA	$(0,2)$	≥ 0	$\mathbb{E}\left[D(y_*) - D(y_0)\right]$	linear	6
smSGD/smSN/smSPP	$(0,2)$	≥ 0	$\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}]$	linear	7
smSGD/smSN/smSPP	$(0,2)$	≥ 0	$\mathbb{E}[f(x_k) - f(x_*)]$	linear	7

Table 3: Summary of the iteration complexity results obtained in this paper. Parameters of the methods: ω (stepsize) and β (momentum term). In all cases, $x_* = \Pi^{\mathbf{B}}_{\mathcal{L}}(x_0)$ is the solution of the best approximation problem. Theorem 3 refers to Cesaro averages: $\hat{x}_k = \frac{1}{k}\sum_{t=0}^{k-1} x_t$. Theorem 6 refers to suboptimality in dual function values (D is the dual function).

where this was shown for the no-momentum methods ($\beta = 0$) and in the special case of the unit stepsize ($\omega = 1$).

Stochastic momentum. We propose a new momentum strategy, which we call *stochastic momentum*. Stochastic momentum is a stochastic (coordinate-wise) approximation of the deterministic momentum, and hence is much less costly, which in some situations leads to computational savings in each iteration. On the other hand, the additional noise introduced this way increases the number of iterations needed for convergence. We analyze the SGD, SN and SPP methods with stochastic momentum, and prove linear convergence rates. We prove that in some settings the overall complexity of SGD with stochastic momentum is better than the overall complexity of SGD with momentum. For instance, this is the case if we consider the randomized Kaczmarz (RK) method as a special case of SGD, and if $\mathbf{A}$ is sparse.

Space for generalizations. We hope that the present work can serve as a starting point for the development of SN, SPP and SDSA methods with momentum for more general classes (beyond special quadratics) of convex and perhaps also nonconvex optimization problems. In such more general settings, however, the symmetry which implies equivalence of these algorithms will break, and hence a different analysis will be needed for each method.

2.5 No need for variance reduction

SGD is arguably one of the most popular algorithms in machine learning. Unfortunately, SGD suffers from slow convergence, which is due to the fact that the variance of the stochastic gradient as an estimator of the gradient does not naturally diminish. For this reason, SGD is typically used with a decreasing stepsize rule, which ensures that the variance converges to zero. However, this has an adverse effect on the convergence rate. For instance, SGD has a sublinear rate even if the function to be minimized is strongly convex. To overcome this problem, a new class of so-called *variance-reduced* methods was developed over the last 2-5 years, including SAG [66], SDCA [68, 62], SVRG/S2GD [29, 32], minibatch SVRG/S2GD [31], and SAGA [12, 11].

Since we assume that the linear system (2) is feasible, it follows that the stochastic gradient vanishes at the optimal point (i.e., $\nabla f_{\mathbf{S}}(x_*) = 0$ for any $\mathbf{S}$). This suggests that additional variance reduction techniques are not necessary since the variance of the stochastic gradient drops to zero as we approach the optimal point x_*. In particular, in our context, SGD with fixed stepsize enjoys linear rate without any variance reduction strategy [42, 23, 64]. Hence, in

this paper we can bypass the development of variance reduction techniques, which allows us to focus on the momentum term.

3 Technical Preliminaries

A general framework for studying consistent linear systems via carefully designed *stochastic reformulations* was recently proposed by Richtárik and Takáč [64]. In particular, given the consistent linear system (2), they provide four reformulations in the form of a stochastic optimization problem, stochastic linear system, stochastic fixed point problem and a stochastic intersection problem. These reformulations are equivalent in the sense that their solutions sets are identical. That is, the set of minimizers of the stochastic optimization problem is equal to the set of solutions of the stochastic linear system and so on. Under a certain assumption, for which the term *exactness* was coined in [64], the solution sets of these reformulations are equal to the solution set of the linear system.

3.1 Stochastic optimization

Stochasticity enters the reformulations via a user defined distribution $\mathcal{D}$ of matrices (all with m rows). In addition, the reformulations utilize a positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ as a parameter, used to define an inner product in $\mathbb{R}^n$ via $\langle x, z\rangle_{\mathbf{B}} := \langle \mathbf{B}x, z\rangle$ and the induced norm $\|x\|_{\mathbf{B}} := (x^\top \mathbf{B}x)^{1/2}$. In particular, the stochastic optimization reformulation (1), i.e., $\min_{x\in\mathbb{R}^n} f(x) := \mathbb{E}\left[f_{\mathbf{S}}(x)\right]$, is defined by setting

$$f_{\mathbf{S}}(x) := \frac{1}{2}\|\mathbf{A}x - b\|^2_{\mathbf{H}} = \frac{1}{2}(\mathbf{A}x - b)^\top \mathbf{H}(\mathbf{A}x - b), \tag{6}$$

where $\mathbf{H}$ is a random symmetric positive semidefinite matrix defined as $\mathbf{H} := \mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top$. By $\dagger$ we denote the Moore-Penrose pseudoinverse.

Hessian and its eigenvalues. Note that the Hessian[4] of $f = \mathbb{E}\left[f_{\mathbf{S}}\right]$ is given by $\nabla^2 f = \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{Z}\right]$, where

$$\mathbf{Z} := \mathbf{A}^\top \mathbf{H}\mathbf{A}. \tag{7}$$

Note that $\nabla^2 f$ and

$$\mathbf{W} := \mathbf{B}^{-1/2}\mathbb{E}\left[\mathbf{Z}\right]\mathbf{B}^{-1/2} \tag{8}$$

have the same spectrum. Matrix $\mathbf{B}$ is symmetric and positive semidefinite (with respect to the standard inner product). Let

$$\mathbf{W} = \mathbf{U}\Lambda\mathbf{U}^\top = \sum_{i=1}^n \lambda_i u_i u_i^\top$$

be the eigenvalue decomposition of $\mathbf{W}$, where $\mathbf{U} = [u_1, \ldots, u_n]$ is an orthonormal matrix of eigenvectors, and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the corresponding eigenvalues. Let $\lambda^+_{\min}$ be the smallest nonzero eigenvalue, and $\lambda_{\max} = \lambda_n$ be the largest eigenvalue. It was shown in [64] that $0 \leq \lambda_i \leq 1$ for all $i \in [n]$.

[4] While the Hessian is not self-adjoint with respect to the standard inner product, it is self-adjoint with respect to the inner product $\langle \mathbf{B}x, y\rangle$ which we use as the canonical inner product in $\mathbb{R}^n$.

Exactness. Note that $f_{\mathbf{S}}$ is a convex quadratic, and that $f_{\mathbf{S}}(x) = 0$ whenever $x \in \mathcal{L} := \{x : \mathbf{A}x = b\}$. However, $f_{\mathbf{S}}$ can be zero also for points x outside of $\mathcal{L}$. Clearly, $f(x)$ is nonnegative, and $f(x) = 0$ for $x \in \mathcal{L}$. However, without further assumptions, the set of minimizers of f can be larger than $\mathcal{L}$. The exactness assumption mentioned above ensures that this does not happen. For necessary and sufficient conditions for exactness, we refer the reader to [64]. Here it suffices to remark that a sufficient condition for exactness is to require $\mathbb{E}[\mathbf{H}]$ to be positive definite. This is easy to see by observing that

$$f(x) = \mathbb{E}[f_{\mathbf{S}}(x)] = \tfrac{1}{2}\|\mathbf{A}x - b\|^2_{\mathbb{E}[\mathbf{H}]}.$$

3.2 Three algorithms for solving the stochastic optimization problem

The authors of [64] consider solving the stochastic optimization problem (1) via stochastic gradient descent (SGD)[5]

$$x_{k+1} = x_k - \omega \nabla f_{\mathbf{S}_k}(x_k), \tag{9}$$

where $\omega > 0$ is a fixed stepsize and $\mathbf{S}_k$ is sampled afresh in each iteration from $\mathcal{D}$. Note that the gradient of $f_{\mathbf{S}}$ with respect to the $\mathbf{B}$ inner product is equal to

$$\nabla f_{\mathbf{S}}(x) \overset{(6)}{=} \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}(\mathbf{A}x - b) = \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}\mathbf{A}(x - x_*) = \mathbf{B}^{-1}\mathbf{Z}(x - x_*), \tag{10}$$

where $\mathbf{Z} := \mathbf{A}^\top \mathbf{H}\mathbf{A}$, and x_* is any vector in $\mathcal{L}$.

They observe that, surprisingly, SGD is in this setting equivalent to several other methods; in particular, to the *stochastic Newton method*[6],

$$x_{k+1} = x_k - \omega(\nabla^2 f_{\mathbf{S}_k}(x_k))^{\dagger_{\mathbf{B}}} \nabla f_{\mathbf{S}_k}(x_k), \tag{11}$$

and to the *stochastic proximal point method*[7]

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ f_{\mathbf{S}_k}(x) + \frac{1-\omega}{2\omega}\|x - x_k\|^2_{\mathbf{B}} \right\}. \tag{12}$$

3.3 Stochastic fixed point problem

The stochastic fixed point problem considered in [64] as one of the four stochastic reformulations has the form

$$x = \mathbb{E}\left[\Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}}}(x)\right], \tag{13}$$

where the expectation is taken with respect to $\mathbf{S} \sim \mathcal{D}$, and where $\Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}}}(x)$ is the projection of x, in the $\mathbf{B}$ norm, onto the sketched system $\mathcal{L}_{\mathbf{S}} = \{x \in \mathbb{R}^n : \mathbf{S}^\top \mathbf{A}x = \mathbf{S}^\top b\}$. An explicit formula for the projection onto $\mathcal{L}$ is given by

$$\Pi^{\mathbf{B}}_{\mathcal{L}}(x) := \arg\min_{x' \in \mathcal{L}} \|x' - x\|_{\mathbf{B}} = x - \mathbf{B}^{-1}\mathbf{A}^\top(\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top)^\dagger(\mathbf{A}x - b); \tag{14}$$

a formula for $\mathcal{L}_{\mathbf{S}}$ is obtained by replacing $\mathbf{A}$ with $\mathbf{S}^\top\mathbf{A}$ everywhere.

The *stochastic fixed point method* (with relaxation parameter $\omega > 0$) for solving (13) is defined by

$$x_{k+1} = \omega \Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}_k}}(x_k) + (1-\omega)x_k. \tag{15}$$

[5] The gradient is computed with respect to the inner product $\langle \mathbf{B}x, y\rangle$.

[6] In this method we take the $\mathbf{B}$-pseudoinverse of the Hessian of $f_{\mathbf{S}_k}$ instead of the classical inverse, as the inverse does not exist. When $\mathbf{B} = \mathbf{I}$, the $\mathbf{B}$ pseudoinverse specializes to the standard Moore-Penrose pseudoinverse.

[7] In this case, the equivalence only works for $0 < \omega \leq 1$.

3.4 Best approximation problem, its dual and SDSA

It was shown in [64] that the above methods converge linearly to $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$; the projection of the initial iterate onto the solution set of the linear system. Hence, besides solving problem (1), they solve the *best approximation problem*

$$\min_{x\in\mathbb{R}^n} P(x) := \tfrac{1}{2}\|x - x_0\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{A}x = b. \tag{16}$$

The Fenchel dual of (16) is the (bounded) unconstrained concave quadratic maximization problem

$$\max_{y\in\mathbb{R}^m} D(y) := (b - \mathbf{A}x_0)^\top y - \tfrac{1}{2}\|\mathbf{A}^\top y\|_{\mathbf{B}^{-1}}^2. \tag{17}$$

Boundedness follows from consistency. It turns out that by varying $\mathbf{A}, \mathbf{B}$ and b (but keeping consistency of the linear system), the dual problem in fact captures *all* bounded unconstrained concave quadratic maximization problems.

In the special case of unit stepsize, method (15) was first proposed by Gower and Richtárik [23] under the name "sketch-and-project method", motivated by the iteration structure which proceeds in two steps: i) replace the set $\mathcal{L} := \{x \in \mathbb{R}^n \ : \ \mathbf{A}x = b\}$ by its *sketched* variant $\mathcal{L}_{\mathbf{S}_k}$, and then project the last iterate x_k onto $\mathcal{L}_{\mathbf{S}_k}$. Analysis in [23] was done under the assumption that $\mathbf{A}$ be of full column rank. This assumption was lifted in [24], and a *duality* theory for the method developed. In particular, for $\omega = 1$, the iterates $\{x_k\}$ arise as images of the iterates $\{y_k\}$ produced by a specific *dual method* for solving (17) under the mapping $\phi : \mathbb{R}^m \mapsto \mathbb{R}^n$ given by

$$\phi(y) := x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y. \tag{18}$$

The dual method—*stochastic dual subspace ascent (SDSA)*—has the form

$$y_{k+1} = y_k + \mathbf{S}_k\lambda_k, \tag{19}$$

where $\mathbf{S}_k$ is in each iteration sampled from $\mathcal{D}$, and λ_k is chosen greedily, maximizing the dual objective D: $\lambda_k \in \arg\max_\lambda D(y_k + \mathbf{S}_k\lambda)$. Such a λ might not be unique, however. SDSA is defined by picking the solution with the smallest (standard Euclidean) norm. This leads to the formula:

$$\lambda_k = \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right).$$

SDSA proceeds by moving in random subspaces spanned by the random columns of $\mathbf{S}_k$. In the special case when $\omega = 1$ and $y_0 = 0$, Gower and Richtárik [24] established the following relationship between the iterates $\{x_k\}$ produced by the primal methods (9), (11), (12), (15) (which are equivalent), and the dual method (19):

$$x_k = \phi(y_k) \overset{(18)}{=} x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k. \tag{20}$$

3.5 Other related work

Variants of the sketch-and-project methods have been recently proposed for solving several other problems. Xiang and Zhang [76] show that the sketch-and-project framework is capable of expressing, as special cases, randomized variants of 16 classical algorithms for solving linear systems. Gower and Richtárik [26, 25] use similar ideas to develop of linearly convergent randomized iterative methods for computing/estimating the inverse and the pseudoinverse of a large matrix, respectively. A limited memory variant of the stochastic block BFGS method for solving the empirical risk minimization problem arising in machine learning was proposed by

Gower et al. [22]. Tu et al. [73] utilize the sketch-and-project framework to show that breaking block locality can accelerate block Gauss-Seidel methods. In addition, they develop an accelerated variant of the method for a specific distribution $\mathcal{D}$. Loizou and Richtárik [38] use the sketch-and-project method to solve the average consensus problem; and Hanzely et al. [28] design new variants of sketch and project methods for the average consensus problem with privacy considerations (see Section 8.3 for more details regarding the average consensus problem).

4 Primal Methods with Momentum

Applied to problem (1), i.e., $\min_{x\in\mathbb{R}^n} f(x) = \mathbb{E}\left[f_{\mathbf{S}}(x)\right]$, the gradient descent method with momentum (also known as the heavy ball method) of Polyak [54, 55] takes the form

$$x_{k+1} = x_k - \omega\nabla f(x_k) + \beta(x_k - x_{k-1}), \tag{21}$$

where $\omega > 0$ is a stepsize and $\beta \geq 0$ is a momentum parameter. Instead of marrying the momentum term with gradient descent, we can marry it with SGD. This leads to SGD with momentum (mSGD), also known as the *stochastic heavy ball method*:

$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \beta(x_k - x_{k-1}). \tag{22}$$

Since SGD is equivalent to SN and SPP, this way we obtain momentum variants of the stochastic Newton (mSN) and stochastic proximal point (mSPP) methods. The method is formally described below:

mSGD / mSN / mSPP

Parameters: Distribution $\mathcal{D}$ from which method samples matrices; positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$; stepsize/relaxation parameter $\omega \in \mathbb{R}$ the heavy ball/momentum parameter β.

Initialize: Choose initial points $x_0, x_1 \in \mathbb{R}^n$

For $k \geq 1$ do

1. Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
2. Set
$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \beta(x_k - x_{k-1})$$

Output: last iterate x_k

To the best of our knowledge, momentum variants of SN and SPP were not considered in the literature before. Moreover, as far as we know, there are no momentum variants of even deterministic variants of (11), (12) and (15), such as incremental or batch Newton method, incremental or batch proximal point method and incremental or batch projection method; not even for a problem formulated differently.

In the rest of this section we state our convergence results for mSGD/mSN/mSPP.

4.1 $L2$ convergence and function values: linear rate

In this section we study L2 convergence of mSGD/mSN/mSPP; that is, we study the convergence of the quantity $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2]$ to zero. We show that for a range of stepsize parameters

$\omega > 0$ and momentum terms $\beta \geq 0$ the method enjoys global linear convergence rate. To the best of our knowledge, these results are the first of their kind for the stochastic heavy ball method. As a corollary of L2 convergence, we obtain convergence of the expected function values.

Theorem 1. *Choose $x_0 = x_1 \in \mathbb{R}^n$. Assume exactness. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of random iterates produced by mSGD/mSN/mSPP. Assume $0 < \omega < 2$ and $\beta \geq 0$ and that the expressions*

$$a_1 := 1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda_{\min}^+, \qquad \text{and} \qquad a_2 := \beta + 2\beta^2 + \omega\beta\lambda_{\max}$$

satisfy $a_1 + a_2 < 1$. Let $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$. Then

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq q^k(1+\delta)\|x_0 - x_*\|_{\mathbf{B}}^2 \tag{23}$$

and

$$\mathbb{E}[f(x_k)] \leq q^k \frac{\lambda_{\max}}{2}(1+\delta)\|x_0 - x_*\|_{\mathbf{B}}^2,$$

where $q = \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta = q - a_1$. Moreover, $a_1 + a_2 \leq q < 1$.

Proof. See Appendix A. □

In the above theorem we obtain a global linear rate. To the best of our knowledge, this is the first time that linear rate is established for a stochastic variant of the heavy ball method (mSGD) in any setting. All existing results are sublinear. These seem to be the first momentum variants of SN and SPP methods.

If we choose $\omega \in (0, 2)$, then the condition $a_1 + a_2 < 1$ is satisfied for all

$$0 \leq \beta < \tfrac{1}{8}\left(-4 + \omega\lambda_{\min}^+ - \omega\lambda_{\max} + \sqrt{(4 - \omega\lambda_{\min}^+ + \omega\lambda_{\max})^2 + 16\omega(2-\omega)\lambda_{\min}^+}\right). \tag{24}$$

If $\beta = 0$, mSGD reduces to SGD analyzed in [64]. In this special case, $q = 1 - \omega(2-\omega)\lambda_{\min}^+$, which is the rate established in [64]. Hence, our result is more general.

Let $q(\beta)$ be the rate as a function of β. Note that since $\beta \geq 0$, we have

$$\begin{aligned} q(\beta) &\geq a_1 + a_2 \\ &= 1 + 4\beta + 4\beta^2 + \omega\beta(\lambda_{\max} - \lambda_{\min}^+) - \omega(2-\omega)\lambda_{\min}^+ \\ &\geq 1 - \omega(2-\omega)\lambda_{\min}^+ = q(0). \end{aligned} \tag{25}$$

Clearly, the lower bound on q is an increasing function of β. Also, for any β the rate is always inferior to that of SGD ($\beta = 0$). It is an open problem whether one can prove a strictly better rate for mSGD than for SGD.

Our next theorem states that $\Pi_{\mathcal{L}}^{\mathbf{B}}(x_k) = x_*$ for all iterations k of mSGD. This invariance is important, as it allows the algorithm to converge to x_*.

Theorem 2. *Let $x_0 = x_1 \in \mathbb{R}^n$ be the starting points of the mSGD method and let $\{x_k\}$ be the random iterates generated by mSGD. Then $\Pi_{\mathcal{L}}^{\mathbf{B}}(x_k) = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$ for all $k \geq 0$.*

Proof. Note that in view of (6), $\nabla f_{\mathbf{S}}(x) = \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}(\mathbf{A}x - b) \in \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$. Since

$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \beta(x_k - x_{k-1}),$$

and since $x_0 = x_1$, it can shown by induction that $x_k \in x_0 + \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$ for all k. However, $\text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$ is the orthogonal complement to $\text{Null}(\mathbf{A})$ in the $\mathbf{B}$-inner product. Since $\mathcal{L}$ is parallel to $\text{Null}(\mathbf{A})$, vectors x_k must have the same $\mathbf{B}$-projection onto $\mathcal{L}$ for all k: $\Pi_{\mathcal{L}}^{\mathbf{B}}(x_0) = x_*$. □

4.2 Cesaro average: sublinear rate without exactness assumption

In this section we present the convergence analysis of the function values computed on the Cesaro average. Again our results are global in nature. To the best of our knowledge are the first results that show $O(1/k)$ convergence of the stochastic heavy ball method. Existing results apply in more general settings at the expense of slower rates. In particular, [77] and [18] get $O(1/\sqrt{k})$ and $O(1/k^{\beta})$ convergence, respectively. When $\beta = 1$, [18] gets $O(1/\log(k))$ rate.

Theorem 3. *Choose $x_0 = x_1$ and let $\{x_k\}_{k=0}^{\infty}$ be the random iterates produced by mSGD/mSN/mSPP, where the momentum parameter $0 \leq \beta < 1$ and relaxation parameter (stepsize) $\omega > 0$ satisfy $\omega + 2\beta < 2$. Let x_* be any vector satisfying $f(x_*) = 0$. If we let $\hat{x}_k = \frac{1}{k}\sum_{t=1}^{k} x_t$, then*

$$\mathbb{E}[f(\hat{x}_k)] \leq \frac{(1-\beta)^2\|x_0 - x_*\|_{\mathbf{B}}^2 + 2\omega\beta f(x_0)}{2\omega(2-2\beta-\omega)k}.$$

Proof. See Appendix B. □

In the special case of $\beta = 0$, the above theorem gives the rate

$$\mathbb{E}[f(\hat{x}_k)] \leq \frac{\|x_0 - x_*\|_{\mathbf{B}}^2}{2\omega(2-\omega)k}.$$

This is the convergence rate for Cesaro averges of the "basic method" (i.e., SGD) established in [64].

Our proof strategy is similar to [20] in which the first global convergence analysis of the (deterministic) heavy ball method was presented. There it was shown that when the objective function has a Lipschitz continuous gradient, the Cesaro averages of the iterates converge to the optimum at a rate of $O(1/k)$. To the best of our knowledge, there are no results in the literature that prove the same rate of convergence in the stochastic case for any class of objective functions.

In [77] the authors analyzed mSGD for general Lipshitz continuous convex objective functions (with bounded variance) and proved the *sublinear* rate $O(1/\sqrt{k})$. In [18], a complexity analysis is provided for the case of quadratic strongly convex smooth coercive functions. A sublinear convergence rate of $O(1/k^{\beta})$, where $\beta \in (0,1)$, was proved. In contrast to our results, where we assume fixed stepsize ω, both papers analyze mSGD with diminishing stepsizes.

4.3 $L1$ convergence: accelerated linear rate

In this section we show that by a proper combination of the relaxation (stepsize) parameter ω and the momentum parameter β, mSGD/mSN/mSPP enjoy an *accelerated* linear convergence rate in mean.

Theorem 4. *Assume exactness. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of random iterates produced by mSGD / mSN / mSPP, started with $x_0, x_1 \in \mathbb{R}^n$ satisfying the relation $x_0 - x_1 \in \text{Range}(\mathbf{B}^{-1}\mathbf{A}^{\top})$, with relaxation parameter (stepsize) $0 < \omega \leq 1/\lambda_{\max}$ and momentum parameter $(1-\sqrt{\omega\lambda_{\min}^{+}})^2 < \beta < 1$. Let $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$. Then there exists constant $C > 0$ such that for all $k \geq 0$ we have*

$$\|\mathbb{E}[x_k - x_*]\|_{\mathbf{B}}^2 \leq \beta^k C.$$

(i) If we choose $\omega = 1$ and $\beta = \left(1 - \sqrt{0.99\lambda_{\min}^{+}}\right)^2$ then $\|\mathbb{E}[x_k - x_*]\|_{\mathbf{B}}^2 \leq \beta^k C$ and the iteration complexity becomes $\tilde{O}\left(\sqrt{1/\lambda_{\min}^{+}}\right)$.

(ii) If we choose $\omega = 1/\lambda_{\max}$ *and* $\beta = \left(1 - \sqrt{\frac{0.99\lambda^+_{\min}}{\lambda_{\max}}}\right)^2$ *then* $\|\mathbb{E}[x_k - x_*]\|^2_{\mathbf{B}} \leq \beta^k C$ *and the iteration complexity becomes* $\tilde{O}\left(\sqrt{\lambda_{\max}/\lambda^+_{\min}}\right)$.

Proof. See Appendix C. □

Note that the convergence factor is precisely equal to the value of the momentum parameter β. Let x be any random vector in $\mathbb{R}^n$ with finite mean $\mathbb{E}[x]$, and $x_* \in \mathbb{R}^n$ is any reference vector (for instance, any solution of $\mathbf{A}x = b$). Then we have the identity (see, for instance [23])

$$\mathbb{E}\left[\|x - x_*\|^2_{\mathbf{B}}\right] = \|\mathbb{E}\left[x - x_*\right]\|^2_{\mathbf{B}} + \mathbb{E}\left[\|x - \mathbb{E}[x]\|^2_{\mathbf{B}}\right]. \tag{26}$$

This means that the quantity $\mathbb{E}\left[\|x - x_*\|^2_{\mathbf{B}}\right]$ appearing in our L2 convergence result (Theorem 1) is larger than $\|\mathbb{E}\left[x - x_*\right]\|^2_{\mathbf{B}}$ appearing in the L1 convergence result (Theorem 4), and hence harder to push to zero. As a corollary, L2 convergence implies L1 convergence. However, note that in Theorem 4 we have established an *accelerated* rate. A similar theorem, also obtaining an accelerated rate in the L1 sense, was established in [64] for an accelerated variant of SGD in the sense of Nesterov.

5 Dual Methods with Momentum

In the previous sections we focused on methods for solving the stochastic optimization problem (1) and the best approximation problem (3). In this section we focus on the dual of the best approximation problem, and propose a momentum variant of SDSA, which we call mSDSA.

Stochastic Dual Subspace Ascent with Momentum (mSDSA)

Parameters: Distribution $\mathcal{D}$ from which method samples matrices; positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$; stepsize/relaxation parameter $\omega \in \mathbb{R}$ the heavy ball/momentum parameter β. SDSA is obtained as a special case of mSDSA for $\beta = 0$.

Initialize: Choose initial points $y_0 = y_1 = 0 \in \mathbb{R}^m$

For $k \geq 1$ do

1. Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
2. Set $\lambda_k = \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right)$
3. Set $y_{k+1} = y_k + \omega \mathbf{S}_k \lambda_k + \beta(y_k - y_{k-1})$

Output: last iterate y_k

5.1 Correspondence between primal and dual methods

In our first result we show that the random iterates of the mSGD/mSN/mSPP methods arise as an affine image of mSDSA under the mapping ϕ defined in (18).

Theorem 5 (Correspondence Between Primal and Dual Methods)**.** *Let* $x_0 = x_1$ *and let* $\{x_k\}$ *be the iterates of mSGD/mSN/mSPP. Let* $y_0 = y_1 = 0$*, and let* $\{y_k\}$ *be the iterates of mSDSA.*

Assume that the methods use the same stepsize $\omega > 0$, momentum parameter $\beta \geq 0$, and the same sequence of random matrices $\mathbf{S}_k$. Then

$$x_k = \phi(y_k) = x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k$$

for all k. That is, the primal iterates arise as affine images of the dual iterates.

Proof. First note that

$$\nabla f_{\mathbf{S}_k}(\phi(y_k)) \overset{(10)}{=} \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(\mathbf{A}\phi(y_k) - b) = -\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_k.$$

We now use this to show that

$$\begin{aligned}
\phi(y_{k+1}) &\overset{(18)}{=} x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_{k+1} \\
&= x_0 + \mathbf{B}^{-1}\mathbf{A}^\top\left[y_k + \omega\mathbf{S}_k\lambda_k + \beta(y_k - y_{k-1})\right] \\
&= \underbrace{x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k}_{\phi(y_k)} + \omega\underbrace{\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_k}_{-\nabla f_{\mathbf{S}_k}(\phi(y_k))} + \beta\mathbf{B}^{-1}\mathbf{A}^\top(y_k - y_{k-1}) \\
&= \phi(y_k) - \omega\nabla f_{\mathbf{S}_k}(\phi(y_k)) + \beta(\mathbf{B}^{-1}\mathbf{A}^\top y_k - \mathbf{B}^{-1}\mathbf{A}^\top y_{k-1}) \\
&\overset{(18)}{=} \phi(y_k) - \omega\nabla f_{\mathbf{S}_k}(\phi(y_k)) + \beta(\phi(y_k) - \phi(y_{k-1})).
\end{aligned}$$

So, the sequence of vectors $\{\phi(y_k)\}$ mSDSA satisfies the same recursion of degree as the sequence $\{x_k\}$ defined by mSGD. It remains to check that the first two elements of both recursions coincide. Indeed, since $y_0 = y_1 = 0$ and $x_0 = x_1$, we have $x_0 = \phi(0) = \phi(y_0)$, and $x_1 = x_0 = \phi(0) = \phi(y_1)$. □

5.2 Convergence

We are now ready to state a linear convergence convergence result describing the behavior of mSDSA in terms of the dual function values $D(y_k)$.

Theorem 6 (Convergence of dual objective)**.** *Choose $y_0 = y_1 \in \mathbb{R}^n$. Assume exactness. Let $\{y_k\}_{k=0}^\infty$ be the sequence of random iterates produced by mSDSA. Assume $0 \leq \omega \leq 2$ and $\beta \geq 0$ and that the expressions*

$$a_1 := 1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda_{\min}^+, \qquad \text{and} \qquad a_2 := \beta + 2\beta^2 + \omega\beta\lambda_{\max}$$

satisfy $a_1 + a_2 < 1$. Let $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$ and let y_* be any dual optimal solution. Then

$$\mathbb{E}[D(y_*) - D(y_k)] \leq q^k(1+\delta)\left[D(y_*) - D(y_0)\right] \tag{27}$$

where $q = \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta = q - a_1$. Moreover, $a_1 + a_2 \leq q < 1$.

Proof. This follows by applying Theorem 1 together with Theorem 5 and the identity $\frac{1}{2}\|x_k - x_0\|_{\mathbf{B}}^2 = D(y_*) - D(y_k)$. □

Note that for $\beta = 0$, mSDSA simplifies to SDSA. Also recall that for unit stepsize ($\omega = 1$), SDSA was analyzed in [23]. In the $\omega = 1$ and $\beta = 0$ case, our result specializes to that established in [23]. Following similar arguments to those in [23], the same rate of convergence can be proved for the duality gap $\mathbb{E}[P(x_k) - D(y_k)]$.

6 Methods with Stochastic Momentum

To motivate *stochastic momentum*, for simplicity fix $\mathbf{B} = \mathbf{I}$, and assume that $\mathbf{S}_k$ is chosen as the jth random unit coordinate vector of $\mathbb{R}^m$ with probability $p_j > 0$. In this case, SGD (9) reduces to the randomized Kaczmarz method for solving the linear system $\mathbf{A}x = b$, first analyzed for $p_j \sim \|\mathbf{A}_{j:}\|^2$ by Strohmer and Vershynin [69].

In this case, mSGD becomes the *randomized Kaczmarz method with momentum* (mRK), and the iteration (22) takes the explicit form

$$x_{k+1} = x_k - \omega \frac{\mathbf{A}_{j:}x_k - b_j}{\|\mathbf{A}_{j:}\|^2}\mathbf{A}_{j:}^\top + \beta(x_k - x_{k-1}).$$

Note that the cost of one iteration of this method is $\mathcal{O}(\|\mathbf{A}_{j:}\|_0 + n)$, where the cardinality term $\|\mathbf{A}_{j:}\|_0$ comes from the stochastic gradient part, and n comes from the momentum part. When $\mathbf{A}$ is sparse, the second term will dominate. Similar considerations apply for many other (but clearly not all) distributions $\mathcal{D}$.

In such circumstances, we propose to replace the expensive-to-compute momentum term by a cheap-to-compute stochastic approximation thereof. In particular, we let i_k be chosen from $[n]$ uniformly at random, and replace $x_k - x_{k-1}$ with $v_{i_k} := e_{i_k}^\top(x_k - x_{k-1})e_{i_k}^\top$, where $e_{i_k} \in \mathbb{R}^n$ is the i_kth unit basis vector in $\mathbb{R}^n$, and β with $n\beta$. Note that v_{i_k} can be computed in $\mathcal{O}(1)$ time. Moreover,

$$\mathbb{E}_{i_k}[n\beta v_{i_k}] = \beta(x_k - x_{k-1}).$$

Hence, we replace the momentum term by an unbiased estimator, which allows us to cut the cost to $\mathcal{O}(\|\mathbf{A}_{j:}\|_0)$.

6.1 Primal methods with stochastic momentum

We now propose a variant of the SGD/SN/SPP methods employing stochastic momentum (smSGD/smSN/smSPP). Since SGD, SN and SPP are equivalent, we will describe the development from the perspective of SGD. In particular, we propose the following method:

$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + n\beta e_{i_k}^\top(x_k - x_{k-1})e_{i_k}. \tag{28}$$

The method is formalize below:

smSGD/smSN/smSPP

Parameters: Distribution $\mathcal{D}$ from which the method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$ the heavy ball/momentum parameter β.

Initialize: Choose initial points $x_1 = x_0 \in \mathbb{R}^n$; set $\mathbf{B} = \mathbf{I} \in \mathbb{R}^{n\times n}$

For $k \geq 1$ do

1. Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
2. Pick $i_k \in [n]$ uniformly at random
3. Set
$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \beta e_{i_k}^\top(x_k - x_{k-1})e_{i_k}$$

Output: last iterate x_k

6.2 Convergence

In the next result we establish L2 linear convergence of smSGD/smSN/smSPP. For this we will require the matrix $\mathbf{B}$ to be equal to the identity matrix.

Theorem 7. *Choose $x_0 = x_1 \in \mathbb{R}^n$. Assume exactness. Let $\mathbf{B} = \mathbf{I}$. Let $\{x_k\}_{k=0}^{\infty}$ be the sequence of random iterates produced by smSGD/smSN/smSPP. Assume $0 < \omega < 2$ and $\beta \geq 0$ and that the expressions*

$$a_1 := 1 + 3\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n} - \left(\omega(2-\omega) + \omega\tfrac{\beta}{n}\right)\lambda_{\min}^{+}, \qquad \text{and} \qquad a_2 := \tfrac{1}{n}(\beta + 2\beta^2 + \omega\beta\lambda_{\max}) \tag{29}$$

satisfy $a_1 + a_2 < 1$. Let $x_* = \Pi_{\mathcal{L}}^{\mathbf{I}}(x_0)$. Then

$$\mathbb{E}[\|x_{k+1} - x_*\|^2] \leq q^k(1+\delta)\|x_0 - x_*\|^2 \tag{30}$$

and $\mathbb{E}[f(x_k)] \leq q^k \frac{\lambda_{\max}}{2}(1+\delta)\|x_0 - x_*\|^2$, where $q := \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta := q - a_1$. Moreover, $a_1 + a_2 \leq q < 1$.

Proof. See Appendix D. □

It is straightforward to see that if we choose $\omega \in (0,2)$, then the condition $a_1 + a_2 < 1$ is satisfied for all β belonging to the interval

$$0 \leq \beta < \tfrac{1}{8}\left(-4 + \omega\lambda_{\min}^{+} - \omega\lambda_{\max} + \sqrt{(4 - \omega\lambda_{\min}^{+} + \omega\lambda_{\max})^2 + 16n\omega(2-\omega)\lambda_{\min}^{+}}\right).$$

The upper bound is similar to that for mSGD/mSN/mSPP; the only difference is an extra factor of n next to the constant 16.

6.3 Momentum versus stochastic momentum

As indicated in the introduction, if we wish to compare mSGD to smSGD used with momentum parameter β, it makes sense to use momentum parameter βn in smSGD. This is because the momentum term in smSGD will then be an unbiased estimator of the deterministic momentum term used in mSGD.

Let $q(\beta)$ be the convergence constant for mSGD with stepsize $\omega = 1$ and an admissible momentum parameter $\beta \geq 0$. Further, let $\bar{a}_1(\beta), \bar{a}_2(\beta), \bar{q}(\beta)$ be the convergence constants for smSGD with stepsize $\omega = 1$ and momentum parameter β. We have

$$\begin{aligned} \bar{q}(\beta n) \geq \bar{a}_1(\beta n) + \bar{a}_2(\beta n) &\overset{(29)}{=} 1 + 4\beta + 4\beta^2 n + \beta(\lambda_{\max} - \lambda_{\min}^{+}) - \lambda_{\min}^{+} \\ &\overset{(25)}{=} a_1(\beta) + a_2(\beta) + 4\beta^2(n-1) \\ &\geq a_1(\beta) + a_2(\beta). \end{aligned}$$

Hence, the lower bound on the rate for smSGD is worse than the lower bound for mSGD.

The same conclusion holds for the convergence rates themselves. Indeed, note that since $\bar{a}_1(\beta n) - a_1(\beta) = 2\beta^2(n-1) \geq 0$ and $\bar{a}_2(\beta n) - a_2(\beta) = 2\beta^2(n-1) \geq 0$, we have

$$\bar{q}(\beta n) = \frac{\bar{a}_1(\beta n) + \sqrt{\bar{a}_1^2(\beta n) + 4\bar{a}_2(\beta n)}}{2} \geq \frac{a_1(\beta) + \sqrt{a_1^2(\beta) + 4a_2(\beta)}}{2} = q(\beta),$$

and hence the rate of mSGD is always better than that of smSGD.

However, the expected cost of a single iteration of mSGD may be significantly larger than that of smSGD. Indeed, let g be the expected cost of evaluating a stochastic gradient. Then we need to compare $\mathcal{O}(g+n)$ (mSGD) against $\mathcal{O}(g)$ (smSGD). If $g \ll n$, then one iteration of smSGD is significantly cheaper than one iteration of mSGD. Let us now compare the total complexity to investigate the trade-off between the rate and cost of stochastic gradient evaluation. Ignoring constants, the total cost of the two methods (cost of a single iteration multiplied by the number of iterations) is:

$$C_{\text{mSGD}}(\beta) := \frac{g+n}{1-q(\beta)} = \frac{g+n}{1-\frac{a_1(\beta)+\sqrt{a_1^2(\beta)+4a_2(\beta)}}{2}}, \tag{31}$$

and

$$C_{\text{smSGD}}(\beta n) := \frac{g}{1-\bar{q}(\beta n)} = \frac{g}{1-\frac{\bar{a}_1(\beta n)+\sqrt{\bar{a}_1^2(\beta n)+4\bar{a}_2(\beta n)}}{2}}. \tag{32}$$

Since

$$q(0) = \bar{q}(0n), \tag{33}$$

and since $q(\beta)$ and $\bar{q}(\beta n)$ are continuous functions of β, then because $g+n > g$, for small enough β we will have $C_{\text{mSGD}}(\beta) > C_{\text{smSGD}}(\beta n)$. In particular, the speedup of smSGD compared to mSGD for $\beta \approx 0$ will be close to

$$\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)} \approx \lim_{\beta' \to +0} \frac{C_{\text{mSGD}}(\beta')}{C_{\text{smSGD}}(\beta' n)} \overset{(31)+(32)+(33)}{=} \frac{g+n}{g} = 1 + \frac{n}{g}.$$

Thus, we have shown the following statement.

Theorem 8. *For small β, the total complexity of smSGD is approximately $1+n/g$ times smaller than the total complexity of mSGD, where n is the number of columns of $\mathbf{A}$, and g is the expected cost of evaluating a stochastic gradient $\nabla f_{\mathbf{S}}(x)$.*

7 Special Cases: Randomized Kaczmarz with Momentum and Randomized Coordinate Descent with Momentum

In Table 4 we specify several special instances of mSGD by choosing distinct combinations of the parameters $\mathcal{D}$ and $\mathbf{B}$. We use e_i to denote the ith unit coordinate vector in $\mathbb{R}^m$, and $\mathbf{I}_{:C}$ for the column submatrix of the $m \times m$ identity matrix indexed by (a random) set C.

The updates for smSGD can be derived by substituting the momentum term $\beta(x_k - x_{k-1})$ with its stochastic variant $n\beta e_{i_k}^\top (x_k - x_{k-1}) e_{i_k}$. We do not aim to be comprehensive. For more details on the possible combinations of the parameters $\mathbf{S}$ and $\mathbf{B}$ we refer the interested reader to Section 3 of [23].

In the rest of this section we present in detail two special cases: the randomized Kaczmarz method with momentum (mRK) and the randomized coordinate descent method with momentum (mRCD). Further, we compare the $L1$ convergence rates (i.e., bounds on $\|\mathbb{E}[x_k] - x_*\|_{\mathbf{B}}^2$) obtained in this paper with rates that can be inferred from known results for their no-momentum variants.

7.1 mRK: randomized Kaczmarz with momentum

We now provide a discussion on mRCD (the method in the first row of Table 4). Let $\mathbf{B} = \mathbf{I}$ and let pick in each iteration the random matrix $\mathbf{S} = e_i$ with probability $p_i = \|\mathbf{A}_{i:}\|^2 / \|\mathbf{A}\|_F^2$. In

Variants of mSGD			
Variant of mSGD	$\mathbf{S}$	$\mathbf{B}$	x_{k+1}
mRK: randomized Kaczmarz with momentum	e_i	$\mathbf{I}$	$x_k - \omega \frac{\mathbf{A}_{i:}x_k - b_i}{\|\mathbf{A}_{i:}\|_2^2}\mathbf{A}_{i:}^\top + \beta(x_k - x_{k-1})$
mRCD = mSDSA: randomized coordinate desc. with momentum	e_i	$\mathbf{A} \succ 0$	$x_k - \omega \frac{(\mathbf{A}_{i:})^\top x_k - b_i}{\mathbf{A}_{ii}}e_i + \beta(x_k - x_{k-1})$
mRBK: randomized block Kaczmarz with momentum	$\mathbf{I}_{:C}$	$\mathbf{I}$	$x_k - \omega \mathbf{A}_{C:}^\top(\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger(\mathbf{A}_{C:}x_k - b_C) + \beta(x_k - x_{k-1})$
mRCN = mSDSA: randomized coordinate Newton descent with momentum	$\mathbf{I}_{:C}$	$\mathbf{A} \succ 0$	$x_k - \omega \mathbf{I}_{:C}(\mathbf{I}_{:C}^\top \mathbf{A}\mathbf{I}_{:C})^\dagger \mathbf{I}_{:C}^\top(\mathbf{A}x_k - b) + \beta(x_k - x_{k-1})$
mRGK: randomized Gaussian Kaczmarz	$N(0, \mathbf{I})$	$\mathbf{I}$	$x_k - \omega \frac{\mathbf{S}^\top(\mathbf{A}x_k - b)}{\|\mathbf{A}^\top \mathbf{S}\|_2^2}\mathbf{A}^\top \mathbf{S} + \beta(x_k - x_{k-1})$
mRCD: randomized coord. descent (least squares)	$\mathbf{A}_{:i}$	$\mathbf{A}^\top \mathbf{A}$	$x_k - \omega \frac{(\mathbf{A}_{:i})^\top(\mathbf{A}x_k - b)}{\|\mathbf{A}_{:i}\|_2^2}e_i + \beta(x_k - x_{k-1})$

Table 4: Selected special cases of mSGD. In the special case of $\mathbf{B} = \mathbf{A}$, mSDSA is directly equivalent to mSGD (this is due to the primal-dual relationship (20); see also Theorem 5). Randomized coordinate Newton (RCN) method was first proposed in [60]; mRCN is its momentum variant. Randomized Gaussian Kaczmarz (RGK) method was first proposed in [23]; mRGK is its momentum variant.

this setup the update rule of the mSGD simplifies to

$$x_{k+1} = x_k - \omega \frac{\mathbf{A}_{i:}x_k - b_i}{\|\mathbf{A}_{i:}\|_2^2}\mathbf{A}_{i:}^\top + \beta(x_k - x_{k-1})$$

and

$$\mathbf{W} \overset{(8)}{=} \mathbf{B}^{-1/2}\mathbf{A}^\top \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\mathbf{H}]\mathbf{A}\mathbf{B}^{-1/2} = \mathbb{E}[\mathbf{A}^\top\mathbf{H}\mathbf{A}]$$
$$= \sum_{i=1}^m p_i \frac{\mathbf{A}_{i:}^\top \mathbf{A}_{i:}}{\|\mathbf{A}_{i:}\|_2^2} = \frac{1}{\|\mathbf{A}\|_F^2}\sum_{i=1}^m \mathbf{A}_{i:}^\top\mathbf{A}_{i:} = \frac{\mathbf{A}^\top\mathbf{A}}{\|\mathbf{A}\|_F^2}. \tag{34}$$

The objective function takes the following form:

$$f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[f_{\mathbf{S}}(x)] = \sum_{i=1}^m p_i f_{\mathbf{S}_i}(x) = \frac{\|\mathbf{A}x-b\|_2^2}{2\|\mathbf{A}\|_F^2}. \tag{35}$$

For $\beta = 0$, this method reduces to the *randomized Kaczmarz method* with relaxation, first analyzed in [64]. If we also have $\omega = 1$, this is equivalent with the *randomized Kaczmarz method* of Strohmer and Vershynin [69]. RK without momentum ($\beta = 0$) and without relaxation ($\omega = 1$) converges with iteration complexity [69, 23, 24] of

$$\tilde{O}(1/\lambda_{\min}^+(\mathbf{W})) = \tilde{O}\left(\frac{\|\mathbf{A}\|_F^2}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}\right). \tag{36}$$

In contrast, based on Theorem 4 we have

- For $\omega = 1$ and $\beta = \left(1-\sqrt{0.99\lambda_{\min}^+}\right)^2 = \left(1-\sqrt{\frac{0.99}{\|\mathbf{A}\|_F^2}\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}\right)^2$, the iteration complexity of the mRK is:
$$\tilde{O}\left(\sqrt{\frac{\|\mathbf{A}\|_F^2}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}}\right).$$

- For $\omega = \|\mathbf{A}\|_F^2/\lambda_{\max}(\mathbf{A}^\top\mathbf{A})$ and $\beta = \left(1-\sqrt{\frac{0.99\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}}\right)^2$ the iteration complexity becomes:
$$\tilde{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}}\right).$$

This is quadratic improvement on the previous best result (36).

Related Work. The Kaczmarz method for solving consistent linear systems was originally introduced by Kaczmarz in 1937 [30]. This classical method selects the rows to project onto in a cyclic manner. In practice, many different selection rules can be adopted. For non-random selection rules (cyclic, greedy, etc) we refer the interested reader to [56, 5, 50, 57, 8]. In this work we are interested in *randomized* variants of the Kaczmarz method, first analyzed by Strohmer and Vershynin [69]. In [69] it was shown that RK converges with a linear convergence rate to the unique solution of a full-rank consistent linear system. This result sparked renewed interest in design of randomized methods for solving linear systems [41, 43, 15, 40, 80, 44, 67]. All existing results on accelerated variants of RK use the Nesterov's approach of acceleration [34, 37, 73, 64]. To the best of our knowledge, no convergence analysis of mRK exists in the literature (Polyak's momentum). Our work fills this gap.

7.2 mRCD: randomized coordinate descent with momentum

We now provide a discussion on the mRCD method (the method in the second row of Table 4). If the matrix $\mathbf{A}$ is positive definite, then we can choose $\mathbf{B} = \mathbf{A}$ and $\mathbf{S} = e_i$ with probability $p_i = \frac{\mathbf{A}_{ii}}{\text{Trace}(\mathbf{A})}$. It is easy to see that $\mathbf{W} = \frac{\mathbf{A}}{\text{Trace}(\mathbf{A})}$. In this case, $\mathbf{W}$ is positive definite and as a result, $\lambda_{\min}^{+}(\mathbf{W}) = \lambda_{\min}(\mathbf{W})$. Moreover, we have

$$f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[f_{\mathbf{S}}(x)] = \sum_{i=1}^{m} p_i f_{\mathbf{S}_i}(x) = \frac{\|\mathbf{A}x - b\|_2^2}{2\text{Trace}(\mathbf{A})}. \tag{37}$$

For $\beta = 0$ and $\omega = 1$ the method is equivalent with *randomized coordinate descent* of Leventhal and Lewis [36], which was shown to converge with iteration complexity

$$\text{Previous best result:} \qquad \tilde{O}\left(\frac{\text{Trace}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}\right). \tag{38}$$

In contrast, following Theorem 4, we can obtain the following L_1 iteration complexity results for mRCD:

- For $\omega = 1$ and $\beta = \left(1 - \sqrt{\frac{0.99}{\text{Trace}(\mathbf{A})}\lambda_{\min}(\mathbf{A})}\right)^2$, the iteration complexity is

$$\tilde{O}\left(\sqrt{\frac{\text{Trace}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}}\right).$$

- For $\omega = \text{Trace}(\mathbf{A})/\lambda_{\max}(\mathbf{A})$ and $\beta = \left(1 - \sqrt{\frac{0.99\lambda_{\min}(\mathbf{A})}{\lambda_{\max}(\mathbf{A})}}\right)^2$ the iteration complexity becomes

$$\tilde{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}}\right).$$

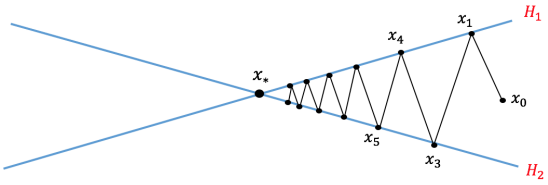
This is quadratic improvement on the previous best result (38).

Related Work. It is known that if $\mathbf{A}$ is positive definite, the popular *randomized Gauss-Seidel* method can be interpreted as randomized coordinate descent (RCD). RCD methods were first analyzed by Lewis and Leventhal in the context of linear systems and least-squares problems [36], and later extended by several authors to more general settings, including smooth convex optimization [47], composite convex optimization [62], and parallel/subspace descent variants [63]. These results were later further extended to handle arbitrary sampling distributions [58, 59, 61, 6] . Accelerated variants of RCD were studied in [34, 16, 1]. For other non-randomized coordinate descent variants and their convergence analysis, we refer the reader to [75, 49, 8]. To the best of our knowledge, mRCD and smRCD have never been analyzed before in any setting.

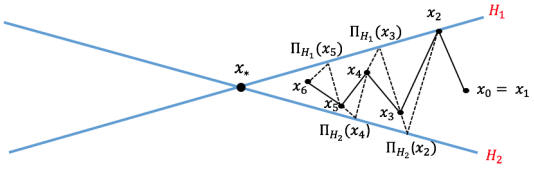
7.3 Visualizing the acceleration mechanism

We devote this section to the graphical illustration of the acceleration mechanism behind momentum. Our goal is to shed more light on how the proposed algorithm works in practice. For simplicity, we illustrate this by comparing RK and mRK.

In Figure 1 we present in a simple $\mathbb{R}^2$ illustration of the difference between the workings of RK and mRK. Our goal is to show graphically how the addition of momentum leads to acceleration. Given iterate x_k, one can think of the update rule of the mRK (22) in two steps:



(a) Randomized Kaczmarz Method [69]



(b) Randomized Kaczmarz Method with Momentum

Figure 1: Graphical interpretation of the *randomized Kaczmarz method* and the *randomized Kaczmarz method with momentum* in a simple example with only two hyperplanes $H_i = \{x \ : \mathbf{A}_{i:}x = b_i\}$ where $i = 1, 2$ and a unique solution x_*.

1. *The Projection:* The projection step corresponds to the first part $x_k - \omega\nabla f_{\mathbf{S}_k}(x_k)$ of the mRK update (22) and it means that the current iterate x_k is projected onto a randomly chosen hyperplane H_i[8]. The value of the stepsize $\omega \in (0, 2)$ defines whether the projection is exact or not. When $\omega = 1$ (no relaxation) the projection is exact, that is the point $\Pi_{H_i}(x_k)$ belongs in the hyperplane H_i. In Figure 1 all projections are exact.

2. *Addition of the momentum term*: The momentum term (right part of the update rule) $\beta(x_k - x_{k-1})$ forces the next iterate x_{k+1} to be closer to the solution x_* than the corresponding point $\Pi_{H_i}(x_k)$. Note also that the vector $x_{k+1} - \Pi_{H_i}(x_k)$ is always parallel to $x_k - x_{k-1}$ for all $k \geq 0$.

Remark 1. *In the example of Figure 1, the performance of mRK is similar to the performance of RK until iterate x_3. After this point, the momentum parameter becomes more effective and the mRK method accelerates. This behavior appears also in our experiments in the next section where we work with matrices with many rows. There we can notice that the momentum parameter seems to become more effective after the first $m + 1$ iterations.*

8 Numerical Experiments

In this section we study the computational behavior of the two proposed algorithms, mSGD and smSGD. In particular, we focus mostly on the evaluation of the performance of mSGD. To highlight the usefulness of smSGD, an empitical verification of Theorem 8 is presented in subsection 8.2. As we have already mentioned, both mSGD and smSGD can be interpreted as sketch-and-project methods (with relaxation), and as a result a comprehensive array of well-known algorithms can be recovered as special cases by varying the main parameters of the methods (check Section 7). In our experiments we focus on the popular special cases of randomized Kaczmarz method (RK) and the randomized coordinate descent method (RCD) without relaxation ($\omega = 1$), and show the practical benefits of the addition of the momentum term[9]. The choice of the stepsize $\omega = 1$ is not arbitrary. Recently, in [64] both relaxed RK and relaxed RCD were analyzed, and it was proved that the quantity $\mathbb{E}\left[\|x_k - x_*\|^2_{\mathbf{B}}\right]$ converges linearly to zero for $\omega \in (0, 2)$, and that the best convergence rate is obtained precisely for $\omega = 1$. Thus the comparison is with the best-in-theory no-momentum variants.

[8] In the plots of Figure 1, the hyperplane of each update is chosen in an alternating fashion for illustration purposes

[9] The experiments were repeated with various values of the main parameters and initializations, and similar results were obtained in all cases.

Note that, convergence analysis of the error $\mathbb{E}\left[\|x_k - x_*\|_{\mathbf{B}}^2\right]$ (L2 convergence) and of the expected function values $\mathbb{E}\left[f(x_k)\right]$ in Theorem 1 shows that mSGD enjoys global non-asymptotic linear convergence rate but not faster than the no-momentum method. The accelerated linear convergence rate has been obtained only in the weak sense (Theorem 4). Nevertheless, in practice as indicated from our experiments, mSGD is faster than its no momentum variant. Note also that in all of the presented experiments the momentum parameters β of the methods are chosen to be positive constants that do not depend on parameters that are not known to the users such as $\lambda_{\min}^+$ and $\lambda_{\max}$.

In comparing the methods with their momentum variants we use both the relative error measure $\|x_k - x_*\|_{\mathbf{B}}^2/\|x_0 - x_*\|_{\mathbf{B}}^2$ and the function values $f(x_k)$[10]. In all implementations, except for the experiments on average consensus (Section 8.3), the starting point is chosen to be $x_0 = 0$. In the case of average consensus the starting point must be the vector with the initial private values of the nodes of the network. All the code for the experiments is written in the Julia programming language. For the horizontal axis we use either the number of iterations or the wall-clock time measured using the tic-toc Julia function.

This section is divided in three main experiments. In the first one we evaluate the performance of the mSGD method in the special cases of mRK and mRCD for solving both synthetic consistent Gaussian systems and consistent linear systems with real matrices. In the second experiment we computationally verify Theorem 8 (comparison between the mSGD and smSGD methods). In the last experiment building upon the recent results of [38] we show how the addition of the momentum accelerates the pairwise randomized gossip (PRG) algorithm for solving the average consensus problem.

Assumptions	No-momentum, $\beta = 0$	Momentum, $\beta \geq 0$	Stochastic Momentum, $\beta \geq 0$
$\mathbf{A}$ general, $\mathbf{B} = \mathbf{I}$	RK	mRK	smRK
$\mathbf{A} \succ 0$, $\mathbf{B} = \mathbf{A}$	RCD	mRCD	smRCD
$\mathbf{A}$ incidence matrix, $\mathbf{B} = \mathbf{I}$	PRG	mPRG	smPRG

Table 5: Abbreviations of the algorithms (special cases of general framework) that we use in the numerical evaluation section. In all methods the random matrices are chosen to be unit coordinate vectors in $\mathbb{R}^m$ ($\mathbf{S} = e_i$). With PRG we denote the Pairwise Randomized Gossip algorithm for solving the average consensus problem first proposed in [4]. Following similar notation with the rest of the paper with mPRG and smPRG we indicate its momentum and stochastic momentum variants respectively.

8.1 Evaluation of mSGD

In this subsection we study the computational behavior of mRK and mRCD when they compared with their no momentum variants for both synthetic and real data.

8.1.1 Synthetic Data

The synthetic data for this comparison is generated as follows[11].

[10] Remember that in our setting we have $f(x_*) = 0$ for the optimal solution x_* of the best approximation problem; thus $f(x) - f(x_*) = f(x)$. The function values $f(x_k)$ refer to function (35) in the case of RK and to function (37) for the RCD. For block variants the objective function of problem (1) has also closed form expression but it can be very difficult to compute. In these cases one can instead evaluate the quantity $\|\mathbf{A}x - b\|_{\mathbf{B}}^2$.

[11] Note that in the first experiment we use Gaussian matrices which by construction are full rank matrices with probability 1 and as a result the consistent linear systems have unique solution. Thus, for any starting

For mRK: All elements of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and of vector $z \in \mathbb{R}^n$ are chosen to be i.i.d $\mathcal{N}(0,1)$. Then the right hand side of the linear system is set to $b = \mathbf{A}z$. With this way the consistency of the linear system with matrix $\mathbf{A}$ and right hand side b is ensured.

For mRCD: A Gaussian matrix $\mathbf{P} \in \mathbb{R}^{m\times n}$ is generated and then matrix $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n\times n}$ is used in the linear system. The vector $z \in \mathbb{R}^n$ is chosen to be i.i.d $\mathcal{N}(0,1)$ and again to ensure consistency of the linear system, the right hand side is set to $b = \mathbf{A}z$.

In particular for the evaluation of mRK we generate Gaussian matrices with $m = 300$ rows and several columns while for the case of mRCD the matrix $\mathbf{P}$ is chosen to be Gaussian with $m = 500$ rows and several columns[12]. Linear systems of these forms were extensively studied [69, 19] and it was shown that the quantity $1/\lambda^+_{\min}$(condition number) can be easily controlled.

For each linear system we run mRK (Figure 2) and mRCD (Figure 3) for several values of momentum parameters β and fixed stepsize $\omega = 1$ and we plot the performance of the methods (average after 10 trials) for both the relative error measure and the function values. Note that for $\beta = 0$ the methods are equivalent with their no-momentum variants RK and RCD respectively.

From Figures 2 and 3 it is clear that the addition of momentum term leads to an improvement in the performance of RK and RCD, respectively. More specifically, from the two figures we observe the following:

- For the well conditioned linear systems ($1/\lambda^+_{\min}$ small) it is known that even the no-momentum variant converges rapidly to the optimal solution. In these cases the benefits of the addition of momentum are not obvious. The momentum term is beneficial for the case where the no-momentum variant ($\beta = 0$) converges slowly, that is when $1/\lambda^+_{\min}$ is large (ill-conditioned linear systems).

- For the case of fixed stepsize $\omega = 1$, the problems with small condition number require smaller momentum parameter β to have faster convergence. Note the first two rows of Figures 2 and 3, where $\beta = 0.3$ or $\beta = 0.4$, are good options.

- For large values of $1/\lambda^+_{\min}$, it seems that the choice of $\beta = 0.5$ is the best. As an example for matrix $\mathbf{A} \in \mathbb{R}^{300\times 280}$ in Figure 2, (where $1/\lambda^+_{\min} = 208,730$), note that to reach relative error 10^{-10}, RK needs around 2 million iterations, while mRK with momentum parameter $\beta = 0.5$ requires only half that many iterations. The acceleration is obvious also in terms of time where in 12 seconds the mRK with momentum parameter $\beta = 0.5$ achieves relative error of the order 10^{-9} and RK requires more than 25 seconds to obtain the same accuracy.

- We observe that both mRK and mRCD, with appropriately chosen momentum parameters $0 < \beta \leq 0.5$, always converge faster than their no-momentum variants, RK and RCD, respectively. This is a smaller momentum parameter than $\beta \approx 0.9$ which is being used extensively with mSGD for training deep neural networks [79, 74, 70].

- In [10] a stochastic power iteration with momentum is proposed for principal component analysis (PCA). There it was demonstrated empirically that a naive application of momentum to the stochastic power iteration does not result in a faster method. To achieve

point x_0, the vector z that used to create the linear system is the solution mSGD converges to. This is not true for general consistent linear systems, with no full-rank matrix. In this case, the solution $x_* = \Pi^{\mathbf{B}}_{\mathcal{L}}(x_0)$ that mSGD converges to is not necessarily equal to z. For this reason, in the evaluation of the relative error measure $\|x_k - x_*\|^2_{\mathbf{B}}/\|x_0 - x_*\|^2_{\mathbf{B}}$, one should be careful and use the value $x_* = x_0 + \mathbf{A}^\dagger(b - \mathbf{A}x_0) \overset{x_0=0}{=} \mathbf{A}^\dagger b$.

[12] RCD converge to the optimal solution only in the case of positive definite matrices. For this reason $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n\times n}$ is used which with probability 1 is a full rank matrix

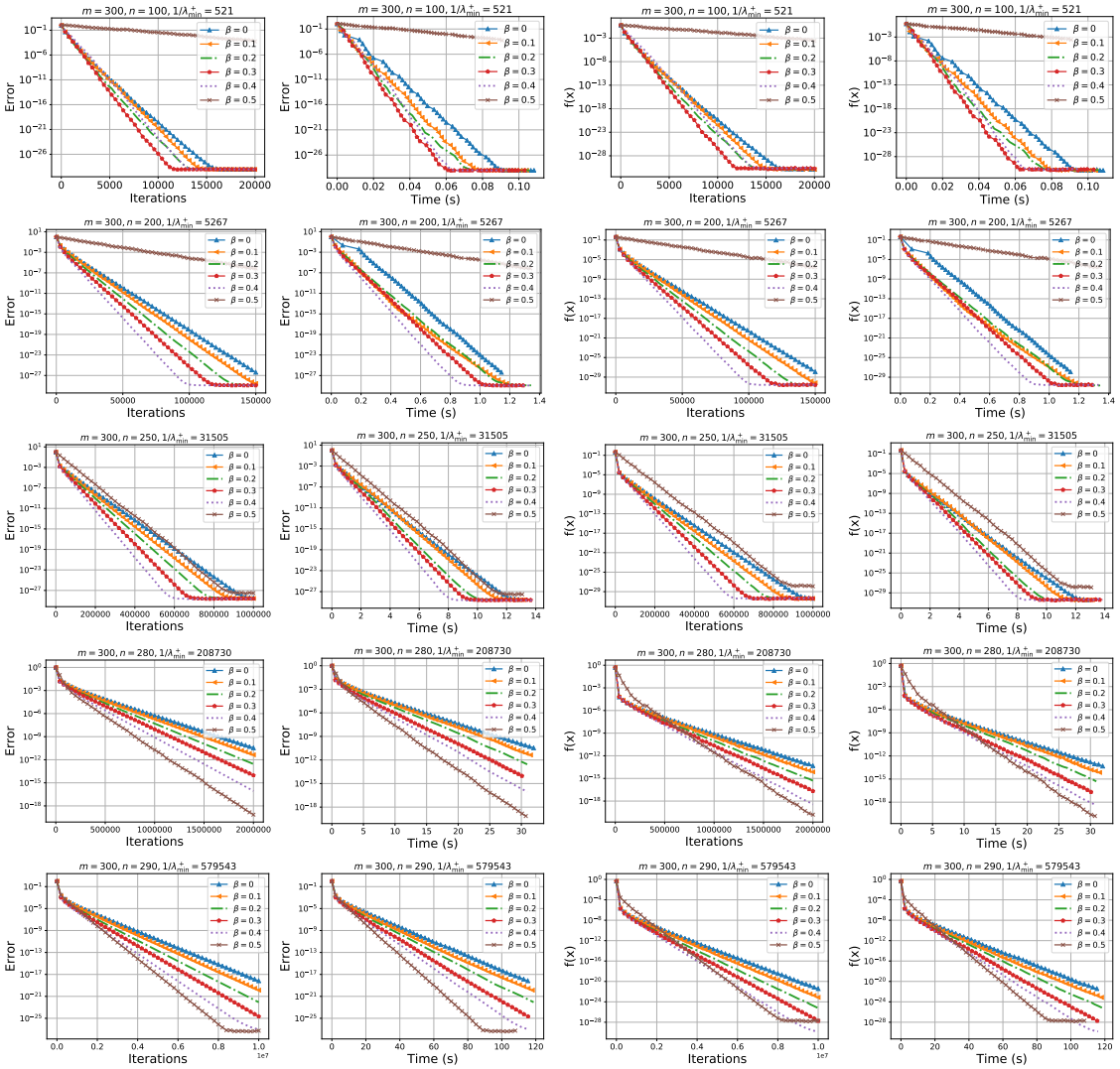


Figure 2: Performance of mRK for fixed stepsize $\omega = 1$ and several momentum parameters β for consistent linear systems with Gaussian matrix $\mathbf{A}$ with $m = 300$ rows and $n = 100, 200, 250, 280, 290$ columns. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. All plots are averaged over 10 trials. The title of each plot indicates the dimensions of the matrix $\mathbf{A}$ and the value of $1/\lambda_{\min}^{+}$. The "Error" on the vertical axis represents the relative error $\|x_k - x_*\|_{\mathbf{B}}^2/\|x_0 - x_*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I},x_0=0}{=} \|x_k - x_*\|^2/\|x_*\|_{\mathbf{B}}^2$ and the function values $f(x_k)$ refer to function (35).

faster convergence, the authors proposed mini-batch and variance-reduction techniques on top of the addition of momentum. In our setting, mere addition of the momentum term to SGD (same is true for special cases such as RK and RCD) leads to empirically faster methods.

8.1.2 Real Data

In the following experiments we test the performance of mRK using real matrices (datasets) from the library of support vector machine problems LIBSVM [7]. Each dataset consists of a

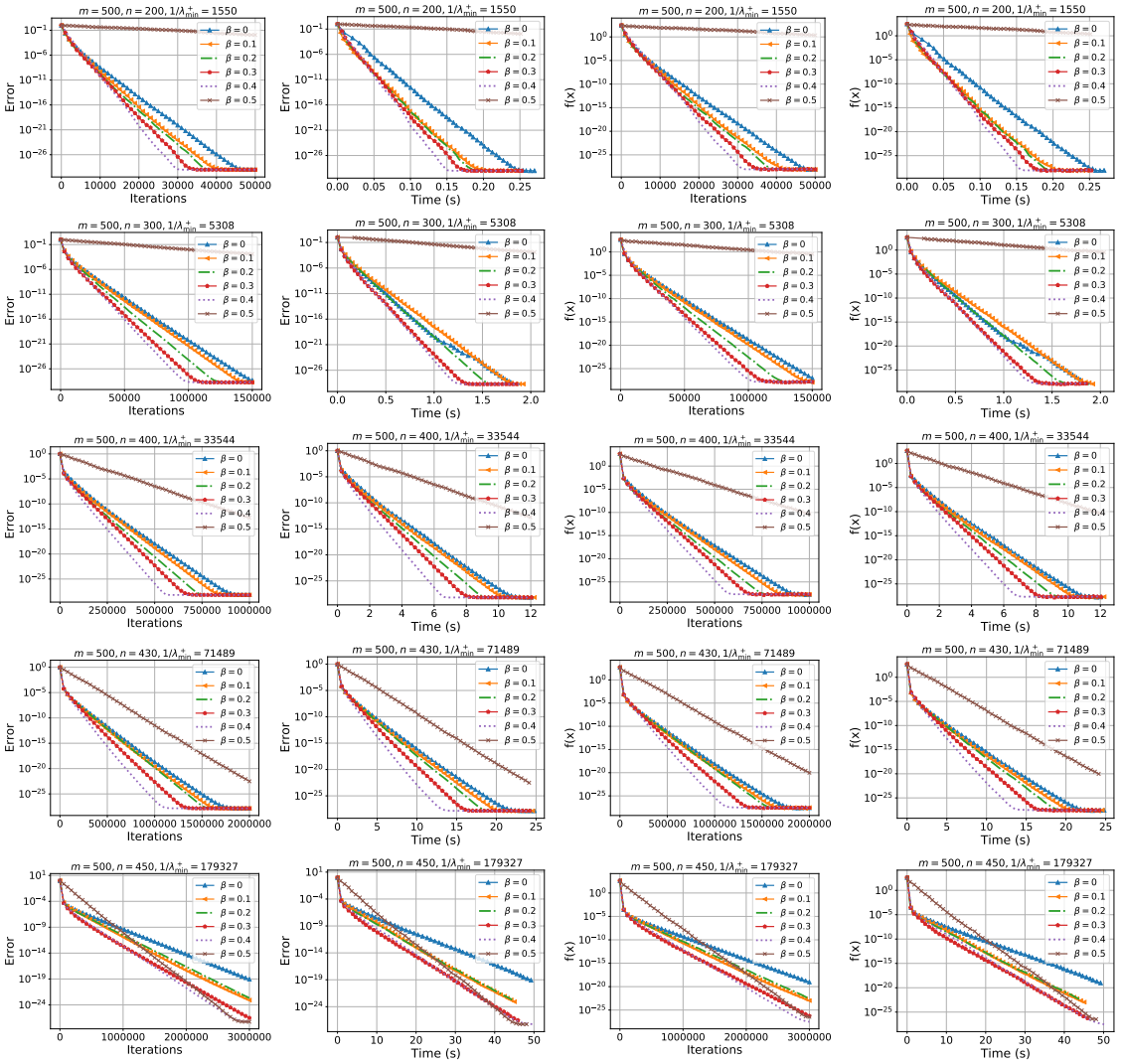


Figure 3: Performance of mRCD for fixed stepsize $\omega = 1$ and several momentum parameters β for consistent linear systems with positive definite matrices $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ where $\mathbf{P} \in \mathbb{R}^{m \times n}$ is Gaussian matrix with $m = 500$ rows and $n = 200, 300, 400, 430, 450$. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. All plots are averaged over 10 trials. The title of each plot indicates the dimensions of the matrix $\mathbf{P}$ and the value of $1/\lambda^+_{\min}$. The "Error" on the vertical axis represents the relative error $\|x_k - x_*\|^2_{\mathbf{B}} / \|x_0 - x_*\|^2_{\mathbf{B}} \overset{\mathbf{B}=\mathbf{A}, x_0=0}{=} \|x_k - x_*\|^2_{\mathbf{A}} / \|x_*\|^2_{\mathbf{A}}$ and the function values $f(x_k)$ refer to function (37).

matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m features and n characteristics) and a vector of labels $b \in \mathbb{R}^m$. In our experiments we choose to use only the matrices of the datasets and ignore the label vector. As before, to ensure consistency of the linear system, we choose a Gaussian vector $z \in \mathbb{R}^n$ and the right hand side of the linear system is set to $b = \mathbf{A}z$. Similarly as in the case of synthetic data, mRK is tested for several values of momentum parameters β and fixed stepsize $\omega = 1$.

In Figure 4 the performance of all methods for both relative error measure $\|x_k - x_*\|^2/\|x_*\|^2_{\mathbf{B}}$ and function values $f(x_k)$ is presented. Note again that $\beta = 0$ represents the baseline RK method. The addition of momentum parameter is again often beneficial and leads to faster convergence. As an example, inspect the plots for the *mushrooms* dataset in Figure 4, where mRK with $\beta = 0.5$ is much faster than the simple RK method in all presented plots, both in terms of iterations and time. In particular, the addition of a momentum parameter leads to visible speedup for the datasets *mushrooms*, *splice*, *a9a* and *ionosphere*. For these datasets the acceleration is obvious in all plots both in terms of relative error and function values. For the datasets *australian*, *gisette* and *madelon* the speedup is less obvious in the plots of the relative error, while for the plots of function values it is not present at all.

8.2 Comparison of momentum & stochastic momentum

In Theorem 8, the total complexities (number of operations needed to achieve a given accuracy) of mSGD and smSGD have been compared and it has been shown that for small momentum parameter β,

$$C_\beta = \frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)} \approx 1 + \frac{n}{g},$$

where C_{mSGD} and C_{smSGD} represent the total costs of the two methods. The goal of this experiment is to show that this relationship holds also in practice.

For this experiment we assume that the non-zeros of matrix $\mathbf{A}$ are not concentrated in certain rows but instead that each row has the same number of non-zero coordinates. We denote by g the number the non-zero elements per row. Having this assumption it can be shown that for the RK method the cost of one projection is equal to $4g$ operations while the cost per iteration of the mRK and of the smRK are $4g + 3n$ and $4g + 1$ respectively. For more details about the cost per iteration of the general mSGD and smSGD check Table 6.

As a first step a Gaussian matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is generated. Then using this matrix several consistent linear systems are obtained as follows. Several values for $g \in [1, n]$ are chosen and for each one of these a matrix $\mathbf{A}_g \in \mathbb{R}^{m\times n}$ with the same elements as $\mathbf{A}$ but with $n - g$ zero coordinates per row is produced. For every matrix $\mathbf{A}_g$, a Gaussian vector $z_g \in \mathbb{R}^n$ is drawn and to ensure consistency of the linear system, the right hand side is set to $b_g = \mathbf{A}_g z$.

We run both mSGD and smSGD with small momentum parameter $\beta = 0.0001$ for solving the linear systems $\mathbf{A}_g x = b_g$ for all selected values of $g \in [1, n]$. The starting point for each run is taken to be $x_0 = 0 \in \mathbb{R}^n$. The methods run until $\epsilon = \|x_k - x_*\| < 0.001$, where $x_* = \Pi_{\mathcal{L}_g}(x_0)$ and $\mathcal{L}_g$ is the solution set of the linear system $\mathbf{A}_g x = b_g$. In each run the number of operations needed to achieve the accuracy ϵ have been counted. For each linear system the average after 10 trials of the value $\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)}$ is computed.

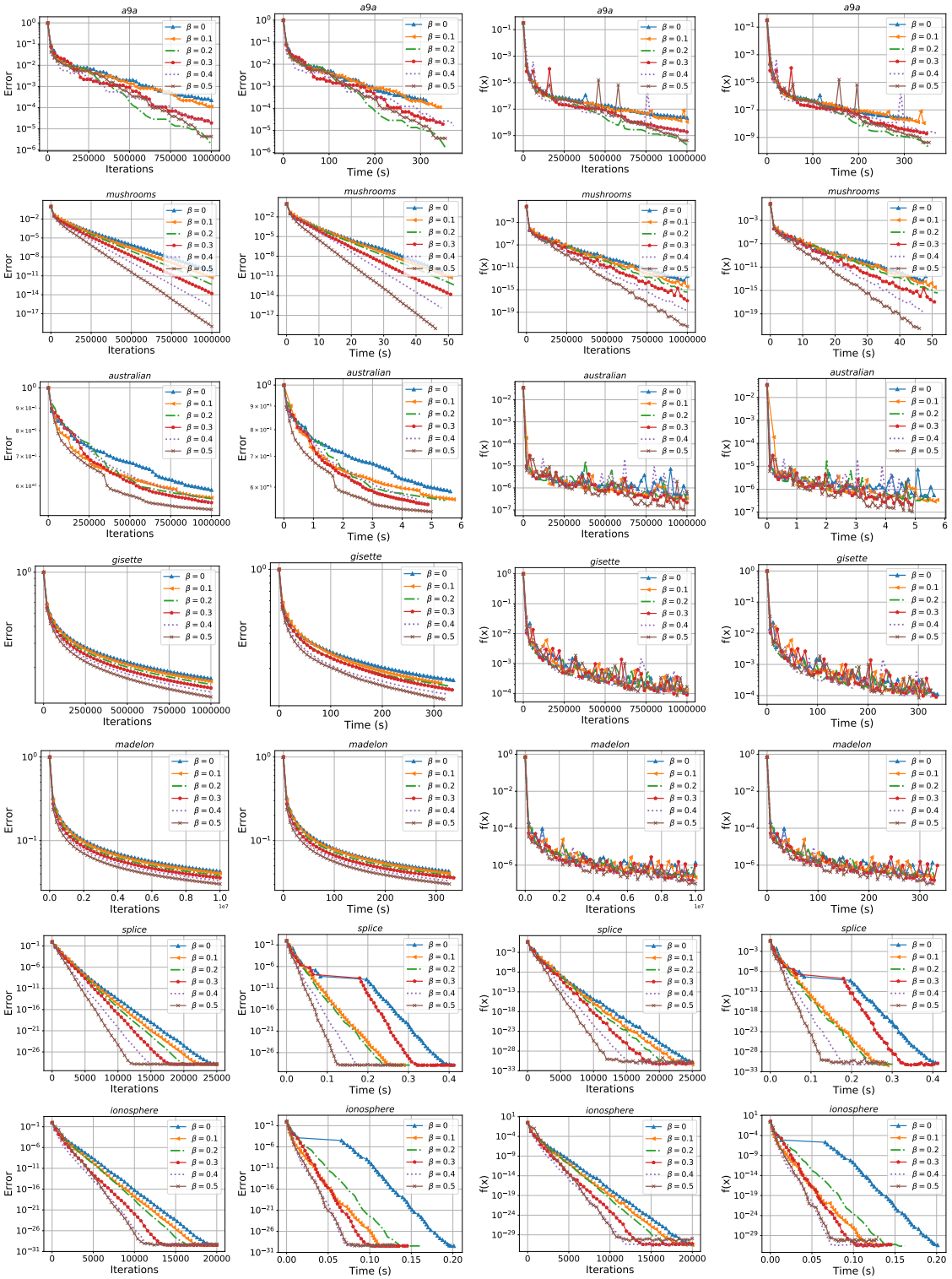
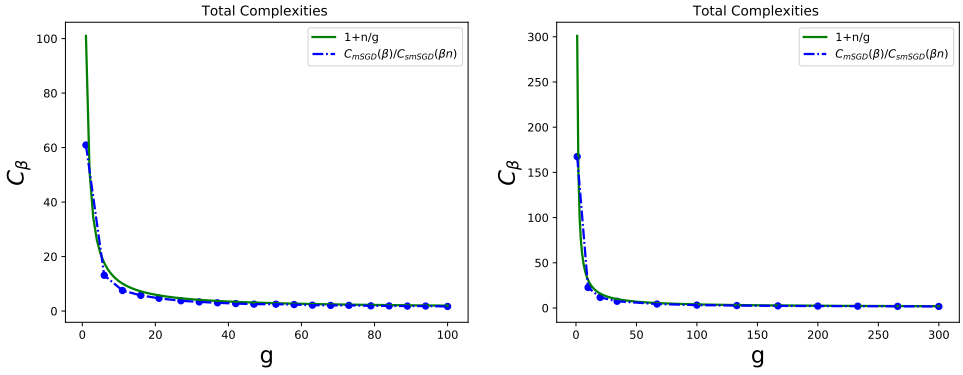


Figure 4: The performance of mRK for several momentum parameters β on real data from LIBSVM [7]. a9a: $(m,n) = (32561, 123)$, mushrooms: $(m,n) = (8124, 112)$, australian: $(m,n) = (690, 14)$, gisette: $(m,n) = (6000, 5000)$, madelon: $(m,n) = (2000, 500)$, splice: $(m,n) = (1000, 60)$, ionosphere: $(m,n) = (351, 34)$. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. The "Error" on the vertical axis represents the relative error $\|x_k - x_*\|_{\mathbf{B}}^2 / \|x_0 - x_*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I}, x_0=0}{=} \|x_k - x_*\|^2 / \|x_*\|_{\mathbf{B}}^2$ and the function values $f(x_k)$ refer to function (35).



(a) $\mathbf{A} \in \mathbb{R}^{200\times 100}$ (b) $\mathbf{A} \in \mathbb{R}^{1000\times 300}$

Figure 5: Comparison of the total complexities of mRK and smRK. The green continuous line denotes the theoretical relationship $1 + \frac{n}{g}$ that we predict in Theorem 8. The blue dotted line shows the ratio of the total complexities $\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)}$ for several linear systems $\mathbf{A}_g x = b_g$ where $g \in [1, n]$. The momentum parameter $\beta = 0.0001$ is used for both methods.

Algorithm	Cost per iteration	Cost per Iteration (RK, mRK, smRK)
Basic Method ($\beta = 0$)	$O(g)$	$4g$
mSGD	$O(g) + O(n) = O(n)$	$4g + 3n$
smSGD	$O(g) + O(1) = O(g)$	$4g + 1$

Table 6: Cost per iteration of the basic, mSGD and smSGD in the general setting and in the special cases of RK,mRK and smRK.

In Figure 5 the actual ratio $\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)}$ and the theoretical approximation $1 + \frac{n}{g}$ are plot and it is shown that they have similar behavior. Thus the theoretical prediction of Theorem 8 is numerically confirmed. In particular in the implementations we use the Gaussian matrices $\mathbf{A} \in \mathbb{R}^{200\times 100}$ and $\mathbf{A} \in \mathbb{R}^{1000\times 300}$.

8.3 Faster method for average consensus

8.3.1 Background

Average consensus (AC) is a fundamental problem in distributed computing and multi-agent systems [13, 4]. Consider a connected undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$, $(|\mathcal{E}| = m)$, where each node $i \in \mathcal{V}$ owns a private value $c_i \in \mathbb{R}$. The goal of the AC problem is each node of the network to compute the average of these private values, $\bar{c} := \frac{1}{n}\sum_i c_i$, via a protocol which allows communication between neighbours only. The problem comes up in many real world applications such as coordination of autonomous agents, estimation, rumour spreading in social networks, PageRank and distributed data fusion on ad-hoc networks and decentralized optimization.

It was shown recently that several randomized methods for solving linear systems can be interpreted as randomized gossip algorithms for solving the AC problem when applied to a special system encoding the underlying network [24, 38]. As we have already explained both basic method [64] and basic method with momentum (this paper) find the solution of the linear system that is closer to the starting point of the algorithms. That is, both methods converge linearly to $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0)$; the projection of the initial iterate onto the solution set of the linear

system and as a result (check Introduction) can be interpreted as methods for solving the best approximation problem (16). In the special case that

1. the linear system in the constraints of (16) is the homogeneous linear system ($\mathbf{A}x = 0$) with matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ being the incidence matrix of the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and

2. the starting point of the method are the initial values of the nodes $x_0 = c$,

it is straightforward to see that the solution of the best approximation problem is a vector with all components equal to the consensus value $\bar{c} := \frac{1}{n}\sum_i c_i$. Under this setting, the famous randomized pairwise gossip algorithm (randomly pick an edge $e \in E$ and replace the private values of its two nodes to their average) that was first proposed and analyzed in [4], is equivalent with the RK method without relaxation ($\omega = 1$) [24, 38].

Remark 2. *In the gossip framework, the condition number of the linear system when RK is used has a simple structure and it depends on the characteristics of the network under study. More specifically, it depends on the number of the edges m and on the Laplacian matrix of the network*[13]*:*

$$\frac{1}{\lambda_{\min}^{+}(\mathbf{W})} \overset{(34)}{=} \frac{1}{\lambda_{\min}^{+}(\mathbf{A}^\top\mathbf{A}/\|\mathbf{A}\|_F^2)} \overset{\|\mathbf{A}\|_F^2=2m}{=} \frac{2m}{\lambda_{\min}^{+}(\mathbf{A}^\top\mathbf{A})} = \frac{2m}{\lambda_{\min}^{+}(\mathbf{L})}, \tag{39}$$

where $\mathbf{L} = \mathbf{A}^\top\mathbf{A}$ is the Laplacian matrix of the network and the quantity $\lambda_{\min}^{+}(\mathbf{L})$ is the very well studied algebraic connectivity of the graph [9].

Remark 3. *The convergence analysis in this paper holds for any consistent linear system $\mathbf{A}x = b$ without any assumption on the rank of the matrix $\mathbf{A}$. The lack of any assumption on the form of matrix $\mathbf{A}$ allows us to solve the homogeneous linear system $\mathbf{A}x = 0$ where $\mathbf{A}$ is the incidence matrix of the network which by construction is rank deficient. More specifically, it can be shown that $\text{rank}(\mathbf{A}) = n - 1$ [38]. Note that many existing methods for solving linear systems make the assumption that the matrix $\mathbf{A}$ of the linear systems is full rank [69, 41, 43] and as a result can not be used to solve the AC problem.*

8.3.2 Numerical Setup

Our goal in this experiment is to show that the addition of the momentum term to the randomized pairwise gossip algorithm (RK in the gossip setting) can lead to faster gossip algorithms and as a result the nodes of the network will converge to the average consensus faster both in number of iterations and in time. We do not intend to analyze the distributed behavior of the method (this is on-going research work). In our implementations we use three of the most popular graph topologies in the literature of wireless sensor networks. These are the line graph, cycle graph and the random geometric graph $G(n, r)$. In practice, $G(n, r)$ consider ideal for modeling wireless sensor networks, because of their particular formulation. In the experiments the 2-dimensional $G(n, r)$ is used which is formed by placing n nodes uniformly at random in a unit square with edges only between nodes that have euclidean distance less than the given radius r. To preserve the connectivity of $G(n, r)$ a radius $r = r(n) = \log(n)/n$ is used [53]. The AC problem is solved for the three aforementioned networks for both $n = 100$ and $n = 200$ number of nodes. We run mRK with several momentum parameters β for 10 trials and we plot their average. Our results are available in Figures 6 and 7.

[13] Matrix $\mathbf{A}$ of the linear system is the incidence matrix of the graph and it is known that the Laplacian matrix is equal to $\mathbf{L} = \mathbf{A}^\top\mathbf{A}$, where $\|\mathbf{A}\|_F^2 = 2m$.

Note that the vector of the initial values of the nodes can be chosen arbitrarily, and the proposed algorithms will find the average of these values. In Figures 6 and 7 the initial value of each node is chosen independently at random from the uniform distribution in the interval $(0, 1)$.

8.3.3 Experimental Results

By observing Figures 6 and 7, it is clear that the addition of the momentum term improves the performance of the popular pairwise randomized gossip (PRG) method [4]. The choice $\beta = 0.4$ as the momentum parameter improves the performance of the vanilla PRG for all networks under study and $\beta = 0.5$ is a good choice for the cases of the cycle and line graph. Note that for networks such as the cycle and line graphs there are known closed form expressions for the algebraic connectivity [9]. Thus, using equation (39), we can compute the exact values of the condition number $1/\lambda_{\min}^{+}$ for these networks. Interestingly, as we can see in Table 7 for $n = 100$ and $n = 200$ (number of nodes), the condition number $1/\lambda_{\min}^{+}$ appearing in the iteration complexity of our methods is not very large. This is in contrast with experimental observations from Section 8.1.1 where it was shown that the choice $\beta = 0.5$ is good for very ill conditioned problems only ($1/\lambda_{\min}^{+}$ very large).

Network	Formula for $\lambda_{\min}^{+}(\mathbf{L})$	$1/\lambda_{\min}^{+}$ for $n = 100$	$1/\lambda_{\min}^{+}$ for $n = 200$
Line	$2\,(1 - \cos(\pi/n))$	1013	4052
Cycle	$2\,(1 - \cos(2\pi/n))$	253	1013

Table 7: Algebraic connectivity of cycle and line graph for $n = 100$ and $n = 200$

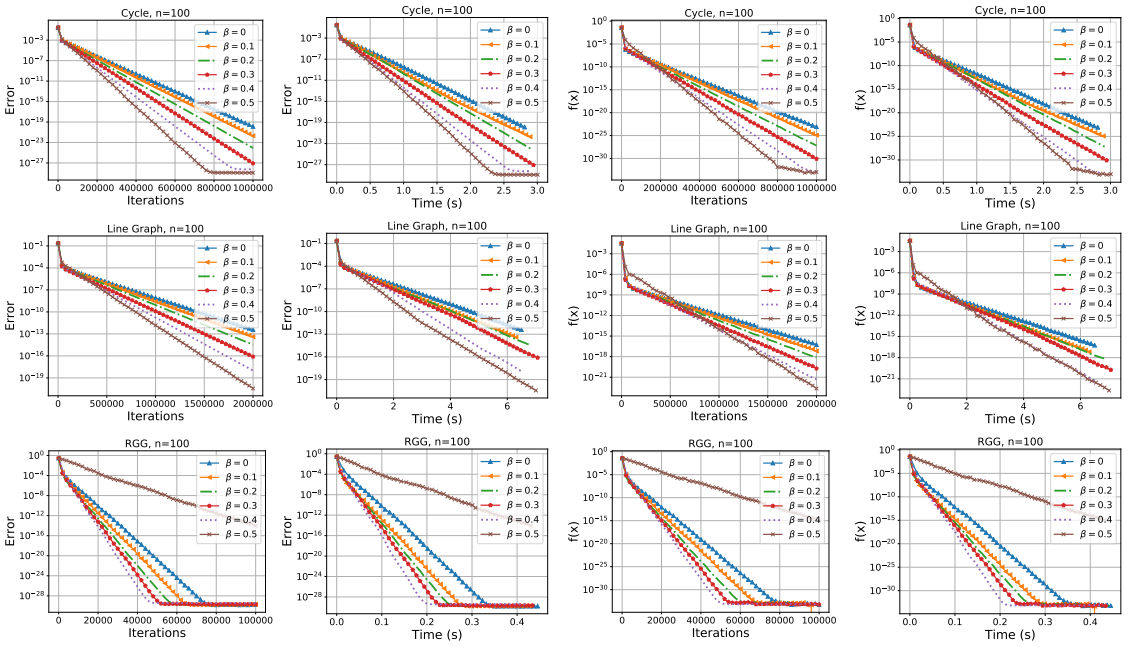


Figure 6: Performance of mPRG for several momentum parameters β for solving the average consensus problem in a cycle graph, line graph and random geometric graph $G(n,r)$ with $n = 100$ nodes. For the $G(n,r)$ to ensure connectivity of the network a radius $r = \sqrt{\log(n)/n}$ is used. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. The "Error" in the vertical axis represents the relative error $\|x_k - x_*\|_{\mathbf{B}}^2 / \|x_0 - x_*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I}, x_0=c}{=} \|x_k - x_*\|^2 / \|c - x_*\|_{\mathbf{B}}^2$ and the function values $f(x_k)$ refer to function (35).

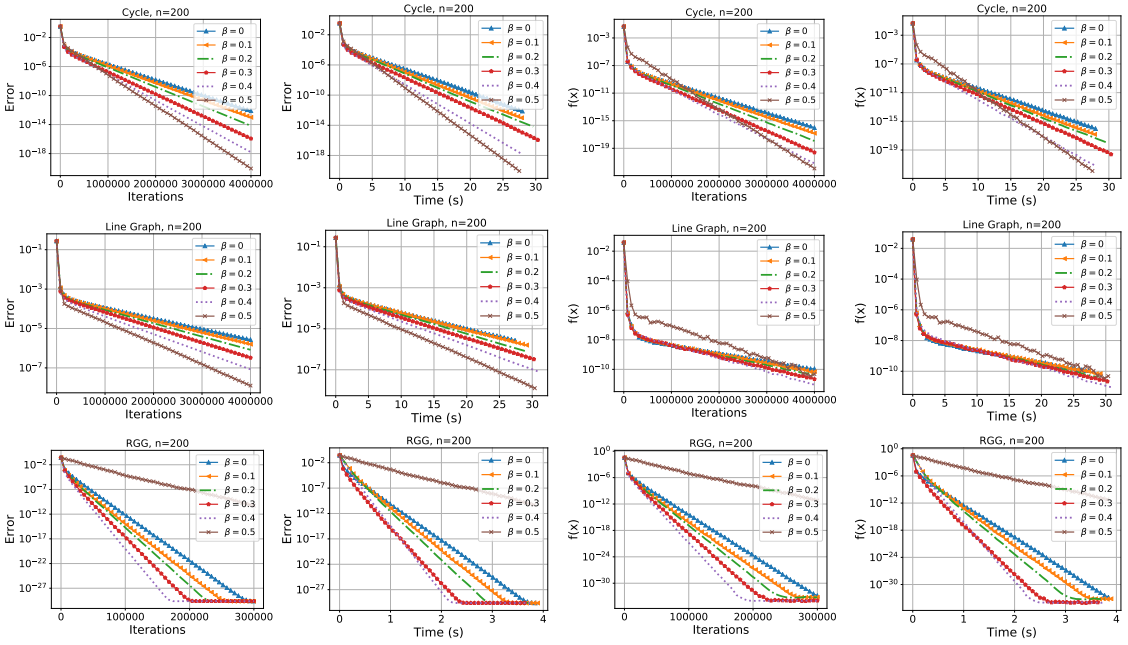


Figure 7: Performance of mPRG for several momentum parameters β for solving the average consensus problem in a cycle graph, line graph and random geometric graph $G(n, r)$ with $n = 200$ nodes. For the $G(n, r)$ to ensure connectivity of the network a radius $r = \sqrt{\log(n)/n}$ is used. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. The "Error" in the vertical axis represents the relative error $\|x_k - x_*\|_{\mathbf{B}}^2 / \|x_0 - x_*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I}, x_0=c}{=} \|x_k - x_*\|^2 / \|c - x_*\|^2$ and the function values $f(x_k)$ refer to function (35).

A Proof of Theorem 1

A.1 Lemmas

We start with a lemma.

Lemma 9. *Fix $F_1 = F_0 \geq 0$ and let $\{F_k\}_{k\geq 0}$ be a sequence of nonnegative real numbers satisfying the relation*

$$F_{k+1} \leq a_1 F_k + a_2 F_{k-1}, \qquad \forall k \geq 1, \tag{40}$$

where $a_2 \geq 0$, $a_1 + a_2 < 1$ and at least one of the coefficients a_1, a_2 is positive. Then the sequence satisfies the relation $F_{k+1} \leq q^k(1+\delta)F_0$ for all $k \geq 1$, where $q = \frac{a_1+\sqrt{a_1^2+4a_2}}{2}$ and $\delta = q - a_1 \geq 0$. Moreover,

$$q \geq a_1 + a_2, \tag{41}$$

with equality if and only if $a_2 = 0$ (in which case $q = a_1$ and $\delta = 0$).

Proof. Choose any $\delta \geq 0$ satisfying $a_2 \leq (a_1 + \delta)\delta$. Adding δF_k to both sides of (40), we get

$$F_{k+1} + \delta F_k \leq (a_1 + \delta)F_k + a_2 F_{k-1} \leq (a_1 + \delta)(F_k + \delta F_{k-1}) = q(F_k + \delta F_{k-1}). \tag{42}$$

We now claim that $\delta = \frac{-a_1+\sqrt{a_1^2+4a_2}}{2}$ satisfies the relations. Non-negativity of δ follows from $a_2 \geq 0$, while the second relation follows from the fact that δ satisfies

$$(a_1 + \delta)\delta - a_2 = 0. \tag{43}$$

Let us now argue that $0 < q < 1$. Nonnegativity of q follows from nonnegativity of a_2. Clearly, as long as $a_2 > 0$, q is positive. If $a_2 = 0$, then $a_1 > 0$ by assumption, which implies that q is positive. The inequality $q < 1$ follows directly from the assumption $a_1 + a_2 < 1$. By unrolling the recurrence (42), we obtain $F_{k+1} \leq F_{k+1} + \delta F_k \leq q^k(F_1 + \delta F_0) = q^k(1 + \delta)F_0$.

Finally, let us establish (42). Noting that $a_1 = q - \delta$, and since in view of (43) we have $a_2 = q\delta$, we conclude that $a_1 + a_2 = q + \delta(q-1) \leq q$, where the inequality follows from $q < 1$. □

The following identities were established in [64]. For completeness, we include different (and somewhat simpler) proofs here.

Lemma 10 ([64])**.** *For all $x \in \mathbb{R}^n$ we have*

$$f_{\mathbf{S}}(x) = \frac{1}{2}\|\nabla f_{\mathbf{S}}(x)\|_{\mathbf{B}}^2. \tag{44}$$

Moreover, if $x_* \in \mathcal{L}$ (i.e., if x_* satisfies $\mathbf{A}x_* = b$), then for all $x \in \mathbb{R}^n$ we have

$$f_{\mathbf{S}}(x) = \frac{1}{2}\langle \nabla f_{\mathbf{S}}(x), x - x_* \rangle_{\mathbf{B}}, \tag{45}$$

and

$$f(x) = \frac{1}{2}\langle \nabla f(x), x - x_* \rangle_{\mathbf{B}}. \tag{46}$$

Proof. In view of (10), and since $\mathbf{Z}\mathbf{B}^{-1}\mathbf{Z} = \mathbf{Z}$ (see [64]), we have

$$\begin{aligned} \|\nabla f_{\mathbf{S}}(x)\|_{\mathbf{B}}^2 &\overset{(10)}{=} \|\mathbf{B}^{-1}\mathbf{Z}(x - x_*)\|_{\mathbf{B}}^2 = (x - x_*)^\top \mathbf{Z}\mathbf{B}^{-1}\mathbf{Z}(x - x_*) = (x - x_*)^\top \mathbf{Z}(x - x_*) \\ &\overset{(7)}{=} (x - x_*)^\top \mathbf{A}^\top \mathbf{H}\mathbf{A}(x - x_*) = (\mathbf{A}x - b)^\top \mathbf{H}(\mathbf{A}x - b) \overset{(6)}{=} 2f_{\mathbf{S}}(x). \end{aligned}$$

Moreover,

$$\begin{aligned}\langle \nabla f_{\mathbf{S}}(x), x - x_* \rangle_{\mathbf{B}} &\overset{(10)}{=} \langle \mathbf{B}^{-1}\mathbf{Z}(x - x_*), x - x_* \rangle_{\mathbf{B}} \\ &= (x - x_*)^\top \mathbf{Z}\mathbf{B}^{-1}\mathbf{B}(x - x_*) \quad = \quad 2f_{\mathbf{S}}(x).\end{aligned}$$

By taking expectations in the last identity with respect to the random matrix $\mathbf{S}$, we get $\langle \nabla f(x), x - x_* \rangle_{\mathbf{B}} = 2f(x)$. □

Lemma 11 ([64]). *For all $x \in \mathbb{R}^n$ and $x_* \in \mathcal{L}$*

$$\lambda_{\min}^{+} f(x) \leq \frac{1}{2}\|\nabla f(x)\|_{\mathbf{B}}^2 \leq \lambda_{\max} f(x) \tag{47}$$

and

$$f(x) \leq \frac{\lambda_{\max}}{2}\|x - x_*\|_{\mathbf{B}}^2. \tag{48}$$

Moreover, if exactness is satisfied, and we let $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x)$, we have

$$\frac{\lambda_{\min}^{+}}{2}\|x - x_*\|_{\mathbf{B}}^2 \leq f(x). \tag{49}$$

A.2 The Proof

First, we decompose

$$\begin{aligned}\|x_{k+1} - x_*\|_{\mathbf{B}}^2 &= \|x_k - \omega \nabla f_{\mathbf{S}_k}(x_k) + \beta(x_k - x_{k-1}) - x_*\|_{\mathbf{B}}^2 \\ &= \underbrace{\|x_k - \omega \nabla f_{\mathbf{S}_k}(x_k) - x_*\|_{\mathbf{B}}^2}_{①} \\ &\quad + \underbrace{2\langle x_k - \omega \nabla f_{\mathbf{S}_k}(x_k) - x_*, \beta(x_k - x_{k-1})\rangle_{\mathbf{B}}}_{②} \\ &\quad + \underbrace{\beta^2 \|x_k - x_{k-1}\|_{\mathbf{B}}^2}_{③}.\end{aligned} \tag{50}$$

We will now analyze the three expressions ①, ②, ③ separately. The first expression can be written as

$$\begin{aligned}① &= \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega\langle x_k - x_*, \nabla f_{\mathbf{S}_k}(x_k)\rangle_{\mathbf{B}} + \omega^2 \|\nabla f_{\mathbf{S}_k}(x_k)\|_{\mathbf{B}}^2 \\ &\overset{(44),(45)}{=} \|x_k - x_*\|_{\mathbf{B}}^2 - 4\omega f_{\mathbf{S}_k}(x_k) + 2\omega^2 f_{\mathbf{S}_k}(x_k) \\ &= \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega(2 - \omega) f_{\mathbf{S}_k}(x_k).\end{aligned} \tag{51}$$

We will now bound the second expression. First, we have

$$\begin{aligned}② &= 2\beta\langle x_k - x_*, x_k - x_{k-1}\rangle_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}} \\ &= 2\beta\langle x_k - x_*, x_k - x_*\rangle_{\mathbf{B}} + 2\beta\langle x_k - x_*, x_* - x_{k-1}\rangle_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}} \\ &= 2\beta\|x_k - x_*\|_{\mathbf{B}}^2 + 2\beta\langle x_k - x_*, x_* - x_{k-1}\rangle_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}}.\end{aligned} \tag{52}$$

Using the fact that for arbitrary vectors $a, b, c \in \mathbb{R}^n$ we have the identity $2\langle a - c, c - b\rangle_{\mathbf{B}} = \|a - b\|_{\mathbf{B}}^2 - \|c - b\|_{\mathbf{B}}^2 - \|a - c\|_{\mathbf{B}}^2$, we obtain

$$2\langle x_k - x_*, x_* - x_{k-1}\rangle_{\mathbf{B}} = \|x_k - x_{k-1}\|_{\mathbf{B}}^2 - \|x_{k-1} - x_*\|_{\mathbf{B}}^2 - \|x_k - x_*\|_{\mathbf{B}}^2.$$

Substituting this into (52) gives

$$\textcircled{2} = \beta\|x_k - x_*\|_{\mathbf{B}}^2 + \beta\|x_k - x_{k-1}\|_{\mathbf{B}}^2 - \beta\|x_{k-1} - x_*\|_{\mathbf{B}}^2 + 2\omega\beta\langle\nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}}. \quad (53)$$

The third expression can be bound as

$$\textcircled{3} = \beta^2\|(x_k - x_*) + (x_* - x_{k-1})\|_{\mathbf{B}}^2 \leq 2\beta^2\|x_k - x_*\|_{\mathbf{B}}^2 + 2\beta^2\|x_{k-1} - x_*\|_{\mathbf{B}}^2. \quad (54)$$

By substituting the bounds (51), (53), (54) into (50) we obtain

$$\begin{aligned}
\|x_{k+1} - x_*\|_{\mathbf{B}}^2 &\leq \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) \\
&\quad +\beta\|x_k - x_*\|_{\mathbf{B}}^2 + \beta\|x_k - x_{k-1}\|_{\mathbf{B}}^2 - \beta\|x_{k-1} - x_*\|_{\mathbf{B}}^2 \\
&\quad +2\omega\beta\langle\nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}} + 2\beta^2\|x_k - x_*\|_{\mathbf{B}}^2 + 2\beta^2\|x_{k-1} - x_*\|_{\mathbf{B}}^2 \\
&\leq (1 + 3\beta + 2\beta^2)\|x_k - x_*\|_{\mathbf{B}}^2 + (\beta + 2\beta^2)\|x_{k-1} - x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) \\
&\quad +2\omega\beta\langle\nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}}.
\end{aligned}$$

Now by first taking expectation with respect to $\mathbf{S}_k$, we obtain:

$$\begin{aligned}
\mathbb{E}_{\mathbf{S}_k}[\|x_{k+1} - x_*\|_{\mathbf{B}}^2] &\leq (1 + 3\beta + 2\beta^2)\|x_k - x_*\|_{\mathbf{B}}^2 + (\beta + 2\beta^2)\|x_{k-1} - x_*\|_{\mathbf{B}}^2 \\
&\quad -2\omega(2-\omega)f(x_k) + 2\omega\beta\langle\nabla f(x_k), x_{k-1} - x_k\rangle_{\mathbf{B}} \\
&\leq (1 + 3\beta + 2\beta^2)\|x_k - x_*\|_{\mathbf{B}}^2 + (\beta + 2\beta^2)\|x_{k-1} - x_*\|_{\mathbf{B}}^2 \\
&\quad -2\omega(2-\omega)f(x_k) + 2\omega\beta(f(x_{k-1}) - f(x_k)) \\
&= (1 + 3\beta + 2\beta^2)\|x_k - x_*\|_{\mathbf{B}}^2 + (\beta + 2\beta^2)\|x_{k-1} - x_*\|_{\mathbf{B}}^2 \\
&\quad -(2\omega(2-\omega) + 2\omega\beta)f(x_k) + 2\omega\beta f(x_{k-1}).
\end{aligned}$$

where in the second step we used the inequality $\langle\nabla f(x_k), x_{k-1} - x_k\rangle \leq f(x_{k-1}) - f(x_k)$ and the fact that $\omega\beta \geq 0$, which follows from the assumptions. We now apply inequalities (48) and (49), obtaining

$$\begin{aligned}
\mathbb{E}_{\mathbf{S}_k}[\|x_{k+1} - x_*\|_{\mathbf{B}}^2] &\leq \underbrace{(1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda_{\min}^+)}_{a_1}\|x_k - x_*\|_{\mathbf{B}}^2 \\
&\quad + \underbrace{(\beta + 2\beta^2 + \omega\beta\lambda_{\max})}_{a_2}\|x_{k-1} - x_*\|_{\mathbf{B}}^2.
\end{aligned}$$

By taking expectation again, and letting $F_k := \mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2]$, we get the relation

$$F_{k+1} \leq a_1 F_k + a_2 F_{k-1}. \quad (55)$$

It suffices to apply Lemma 9 to the relation (55). The conditions of the lemma are satisfied. Indeed, $a_2 \geq 0$, and if $a_2 = 0$, then $\beta = 0$ and hence $a_1 = 1 - \omega(2-\omega)\lambda_{\min}^+ > 0$. The condition $a_1 + a_2 < 1$ holds by assumption.

The convergence result in function values, $\mathbb{E}[f(x_k)]$, follows as a corollary by applying inequality (48) to (27).

B Proof of Theorem 3

Let $p_t = \frac{\beta}{1-\beta}(x_t - x_{t-1})$ and $d_t = \|x_t + p_t - x_*\|_{\mathbf{B}}^2$. In view of (22), we can write

$$x_{t+1} + p_{t+1} = x_t + p_t - \frac{\omega}{1-\beta}\nabla f_{\mathbf{S}_t}(x_t),$$

and therefore

$$\begin{aligned} d_{t+1} &= \left\| x_t + p_t - \frac{\omega}{1-\beta}\nabla f_{\mathbf{S}_t}(x_t) - x_* \right\|_{\mathbf{B}}^2 \\ &= d_t - 2\frac{\omega}{1-\beta}\langle x_t + p_t - x_*, \nabla f_{\mathbf{S}_t}(x_t)\rangle_{\mathbf{B}} + \frac{\omega^2}{(1-\beta)^2}\|\nabla f_{\mathbf{S}_t}(x_t)\|_{\mathbf{B}}^2 \\ &= d_t - \frac{2\omega}{1-\beta}\langle x_t - x_*, \nabla f_{\mathbf{S}_t}(x_t)\rangle_{\mathbf{B}} - \frac{2\omega\beta}{(1-\beta)^2}\langle x_t - x_{t-1}, \nabla f_{\mathbf{S}_t}(x_t)\rangle_{\mathbf{B}} \\ &\quad + \frac{\omega^2}{(1-\beta)^2}\|\nabla f_{\mathbf{S}_t}(x_t)\|_{\mathbf{B}}^2. \end{aligned}$$

Taking expectation with respect to the random matrix $\mathbf{S}_t$ we obtain:

$$\begin{aligned} \mathbb{E}_{\mathbf{S}_t}[d_{t+1}] &= \mathbb{E}_{\mathbf{S}_t}[d_t] - \frac{2\omega}{1-\beta}\langle x_t - x_*, \nabla f(x_t)\rangle_{\mathbf{B}} - \frac{2\omega\beta}{(1-\beta)^2}\langle x_t - x_{t-1}, \nabla f(x_t)\rangle_{\mathbf{B}} \\ &\quad + \frac{\omega^2}{(1-\beta)^2}2f(x_t) \\ &\overset{(46)}{=} \mathbb{E}_{\mathbf{S}_t}[d_t] - \frac{4\omega}{1-\beta}f(x_t) - \frac{2\omega\beta}{(1-\beta)^2}\langle x_t - x_{t-1}, \nabla f(x_t)\rangle_{\mathbf{B}} + \frac{\omega^2}{(1-\beta)^2}2f(x_t) \\ &\leq \mathbb{E}_{\mathbf{S}_t}[d_t] - \frac{4\omega}{1-\beta}f(x_t) - \frac{2\omega\beta}{(1-\beta)^2}[f(x_t) - f(x_{t-1})] + \frac{\omega^2}{(1-\beta)^2}2f(x_t) \\ &= \mathbb{E}_{\mathbf{S}_t}[d_t] + \left[-\frac{4\omega}{1-\beta} - \frac{2\omega\beta}{(1-\beta)^2} + \frac{2\omega^2}{(1-\beta)^2}\right] f(x_t) + \frac{2\omega\beta}{(1-\beta)^2}f(x_{t-1}), \end{aligned}$$

where the inequality follows from convexity of f. After rearranging the terms we get

$$\mathbb{E}_{\mathbf{S}_t}[d_{t+1}] + \frac{2\omega\beta}{(1-\beta)^2}f(x_t) + \alpha f(x_t) \leq \mathbb{E}_{\mathbf{S}_t}[d_t] + \frac{2\omega\beta}{(1-\beta)^2}f(x_{t-1}),$$

where $\alpha = \frac{4\omega}{1-\beta} - \frac{2\omega^2}{(1-\beta)^2} > 0$. Taking expectations again and using the tower property, we get

$$\theta_{t+1} + \alpha\mathbb{E}[f(x_t)] \leq \theta_t, \qquad t = 1, 2, \ldots, \tag{56}$$

where $\theta_t = \mathbb{E}[d_t] + \frac{2\omega\beta}{(1-\beta)^2}\mathbb{E}[f(x_{t-1})]$. By summing up (56) for $t = 1, \ldots, k$ we get

$$\sum_{t=1}^{k}\mathbb{E}[f(x_t)] \leq \frac{\theta_1 - \theta_{k-1}}{\alpha} \leq \frac{\theta_1}{\alpha}. \tag{57}$$

Finally, using Jensen's inequality, we get

$$\mathbb{E}[f(\hat{x}_k)] = \mathbb{E}\left[f\left(\frac{1}{k}\sum_{t=1}^{k}x_t\right)\right] \leq \mathbb{E}\left[\frac{1}{k}\sum_{t=1}^{k}f(x_t)\right] = \frac{1}{k}\sum_{t=1}^{k}\mathbb{E}[f(x_t)] \overset{(57)}{\leq} \frac{\theta_1}{\alpha k}.$$

It remains to note that $\theta_1 = \|x_0 - x_*\|_{\mathbf{B}}^2 + \frac{2\omega\beta}{(1-\beta)^2}f(x_0)$.

C Proof of Theorem 4

In the proof of Theorem 4 the following two lemmas are used.

Lemma 12 ([64]). *Assume exactness. Let $x \in \mathbb{R}^n$ and $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x)$. If $\lambda_i = 0$, then $u_i^\top \mathbf{B}^{1/2}(x - x_*) = 0$.*

Lemma 13 ([14, 17]). *Consider the second degree linear homogeneous recurrence relation:*

$$r_{k+1} = a_1 r_k + a_2 r_{k-1} \tag{58}$$

with initial conditions $r_0, r_1 \in \mathbb{R}$. Assume that the constant coefficients a_1 and a_2 satisfy the inequality $a_1^2 + 4a_2 < 0$ (the roots of the characteristic equation $t^2 - a_1 t - a_2 = 0$ are imaginary). Then there are complex constants C_0 and C_1 (depending on the initial conditions r_0 and r_1) such that:

$$r_k = 2M^k(C_0 \cos(\theta k) + C_1 \sin(\theta k))$$

where $M = \left(\sqrt{\frac{a_1^2}{4} + \frac{(-a_1^2 - 4a_2)}{4}}\right) = \sqrt{-a_2}$ and θ is such that $a_1 = 2M\cos(\theta)$ and $\sqrt{-a_1^2 - 4a_2} = 2M\sin(\theta)$.

We can now turn to the proof of Theorem 4. Plugging in the expression for the stochastic gradient, mSGD can be written in the form

$$\begin{aligned} x_{k+1} &= x_k - \omega \nabla f_{\mathbf{S}_k}(x_k) + \beta(x_k - x_{k-1}) \\ &\overset{(10)}{=} x_k - \omega \mathbf{B}^{-1}\mathbf{Z}_k(x_k - x_*) + \beta(x_k - x_{k-1}). \end{aligned} \tag{59}$$

Subtracting x_* from both sides of (59), we get

$$\begin{aligned} x_{k+1} - x_* &= (\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*) + \beta(x_k - x_* + x_* - x_{k-1}) \\ &= \left((1+\beta)\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k\right)(x_k - x_*) - \beta(x_{k-1} - x_*). \end{aligned}$$

Multiplying the last identity from the left by $\mathbf{B}^{1/2}$, we get

$$\mathbf{B}^{1/2}(x_{k+1} - x_*) = \left((1+\beta)\mathbf{I} - \omega \mathbf{B}^{-1/2}\mathbf{Z}_k\mathbf{B}^{-1/2}\right)\mathbf{B}^{1/2}(x_k - x_*) - \beta \mathbf{B}^{1/2}(x_{k-1} - x_*).$$

Taking expectations, conditioned on x_k (that is, the expectation is with respect to $\mathbf{S}_k$):

$$\mathbf{B}^{1/2}\mathbb{E}[x_{k+1} - x_* \mid x_k] = \left((1+\beta)\mathbf{I} - \omega \mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}\right)\mathbf{B}^{1/2}(x_k - x_*) - \beta \mathbf{B}^{1/2}(x_{k-1} - x_*) \tag{60}$$

Taking expectations again, and using the tower property, we get

$$\begin{aligned} \mathbf{B}^{1/2}\mathbb{E}[x_{k+1} - x_*] &= \mathbf{B}^{1/2}\mathbb{E}\left[\mathbb{E}[x_{k+1} - x_* \mid x_k]\right] \\ &\overset{(60)}{=} \left((1+\beta)\mathbf{I} - \omega \mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}\right)\mathbf{B}^{1/2}\mathbb{E}[x_k - x_*] - \beta \mathbf{B}^{1/2}\mathbb{E}[x_{k-1} - x_*]. \end{aligned}$$

Plugging the eigenvalue decomposition $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ of the matrix $\mathbf{W} = \mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}$ into the above, and multiplying both sides from the left by $\mathbf{U}^\top$, we obtain

$$\mathbf{U}^\top \mathbf{B}^{1/2}\mathbb{E}[x_{k+1} - x_*] = \mathbf{U}^\top \left((1+\beta)\mathbf{I} - \omega \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top\right)\mathbf{B}^{1/2}\mathbb{E}[x_k - x_*] - \beta \mathbf{U}^\top \mathbf{B}^{1/2}\mathbb{E}[x_{k-1} - x_*]. \tag{61}$$

Let us define $s_k := \mathbf{U}^\top \mathbf{B}^{1/2}\mathbb{E}[x_k - x_*] \in \mathbb{R}^n$. Then relation (61) takes the form of the recursion

$$s_{k+1} = [(1+\beta)\mathbf{I} - \omega \mathbf{\Lambda}]s_k - \beta s_{k-1},$$

which can be written in a coordinate-by-coordinate form as follows:

$$s_{k+1}^i = [(1+\beta) - \omega\lambda_i]s_k^i - \beta s_{k-1}^i \quad \text{for all} \quad i = 1, 2, 3, ..., n, \tag{62}$$

where s_k^i indicates the ith coordinate of s_k.

We will now fix i and analyze recursion (62) using Lemma 13. Note that (62) is a second degree linear homogeneous recurrence relation of the form (58) with $a_1 = 1 + \beta - \omega\lambda_i$ and $a_2 = -\beta$. Recall that $0 \le \lambda_i \le 1$ for all i. Since we assume that $0 < \omega \le 1/\lambda_{\max}$, we know that $0 \le \omega\lambda_i \le 1$ for all i. We now consider two cases:

1. $\lambda_i = 0$.

   In this case, (62) takes the form:

   $$s_{k+1}^i = (1+\beta)s_k^i - \beta s_{k-1}^i. \tag{63}$$

   Applying Theorem 2, we know that $x_* = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_0) = \Pi_{\mathcal{L}}^{\mathbf{B}}(x_1)$. Using Lemma 12 twice, once for $x = x_0$ and then for $x = x_1$, we observe that $s_0^i = u_i^\top \mathbf{B}^{1/2}(x_0 - x_*) = 0$ and $s_1^i = u_i^\top \mathbf{B}^{1/2}(x_1 - x_*) = 0$. Finally, in view of (63) we conclude that

   $$s_k^i = 0 \quad \text{for all} \quad k \ge 0. \tag{64}$$

2. $\lambda_i > 0$.

   Since $0 < \omega\lambda_i \le 1$ and $\beta \ge 0$, we have $1 + \beta - \omega\lambda_i \ge 0$ and hence

   $$a_1^2 + 4a_2 = (1 + \beta - \omega\lambda_i)^2 - 4\beta \le (1 + \beta - \omega\lambda_{\min}^+)^2 - 4\beta < 0,$$

   where the last inequality can be shown to hold[14] for $(1 - \sqrt{\omega\lambda_{\min}^+})^2 < \beta < 1$. Applying Lemma 13 the following bound can be deduced

   $$s_k^i \;=\; 2(-a_2)^{k/2}(C_0\cos(\theta k) + C_1\sin(\theta k)) \;\le\; 2\beta^{k/2}P_i, \tag{65}$$

   where P_i is a constant depending on the initial conditions (we can simply choose $P_i = |C_0| + |C_1|$).

Now putting the two cases together, for all $k \ge 0$ we have

$$\begin{aligned}
\|\mathbb{E}[x_k - x_*]\|_{\mathbf{B}}^2 &= \mathbb{E}[x_k - x_*]^\top \mathbf{B}\mathbb{E}[x_k - x_*] \;=\; \mathbb{E}[x_k - x_*]\mathbf{B}^{1/2}\mathbf{U}\mathbf{U}^\top\mathbf{B}^{1/2}\mathbb{E}[x_k - x_*] \\
&= \|\mathbf{U}^\top\mathbf{B}^{1/2}\mathbb{E}[x_k - x_*]\|_2^2 \;=\; \|s_k\|^2 \;=\; \sum_{i=1}^n (s_k^i)^2 \\
&= \sum_{i:\lambda_i=0}(s_k^i)^2 + \sum_{i:\lambda_i>0}(s_k^i)^2 \overset{(64)}{=} \sum_{i:\lambda_i>0}(s_k^i)^2 \\
&\overset{(65)}{\le} \sum_{i:\lambda_i>0} 4\beta^k P_i^2 \\
&= \beta^k C,
\end{aligned}$$

where $C = 4\sum_{i:\lambda_i>0} P_i^2$.

[14] The lower bound on β is tight. However, the upper bound is not. However, we do not care much about the regime of large β as β is the convergence rate, and hence is only interesting if smaller than 1.

D Proof of Theorem 7

The proof follows a similar pattern to that of Theorem 1. However, stochasticity in the momentum term introduces an additional layer of complexity, which we shall tackle by utilizing a more involved version of the tower property.

For simplicity, let $i = i_k$ and $r_k^i := e_i^\top(x_k - x_{k-1})e_i$. First, we decompose

$$\begin{aligned}\|x_{k+1} - x_*\|^2 &= \|x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \beta r_k^i - x_*\|^2 \\ &= \|x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*\|^2 + 2\langle x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*, \beta r_k^i\rangle + \beta^2\|r_k^i\|^2. \quad (66)\end{aligned}$$

We shall use the tower property in the form

$$\mathbb{E}[\mathbb{E}[\mathbb{E}[X \mid x_k, \mathbf{S}_k] \mid x_k]] = \mathbb{E}[X], \quad (67)$$

where X is some random variable. We shall perform the three expectations in order, from the innermost to the outermost. Applying the inner expectation to the identity (66), we get

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k, \mathbf{S}_k] &= \underbrace{\mathbb{E}[\|x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*\|^2 \mid x_k, \mathbf{S}_k]}_{①} \\ &\quad + \underbrace{\mathbb{E}[2\langle x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*, \beta r_k^i\rangle \mid x_k, \mathbf{S}_k]}_{②} \\ &\quad + \underbrace{\mathbb{E}[\beta^2\|r_k^i\|^2 \mid x_k, \mathbf{S}_k]}_{③}. \quad (68)\end{aligned}$$

We will now analyze the three expressions ①, ②, ③ separately. The first expression is constant under the expectation, and hence we can write

$$\begin{aligned}① &= \|x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*\|^2 \\ &= \|x_k - x_*\|^2 - 2\omega\langle x_k - x_*, \nabla f_{\mathbf{S}_k}(x_k)\rangle + \omega^2\|\nabla f_{\mathbf{S}_k}(x_k)\|^2 \\ &\overset{(44)+(45)}{=} \|x_k - x_*\|^2 - 4\omega f_{\mathbf{S}_k}(x_k) + 2\omega^2 f_{\mathbf{S}_k}(x_k) \\ &= \|x_k - x_*\|^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k). \quad (69)\end{aligned}$$

We will now bound the second expression. Using the identity

$$\mathbb{E}[r_k^i \mid x_k, \mathbf{S}_k] = \mathbb{E}_i[r_k^i] = \sum_{i=1}^n \frac{1}{n} r_k^i = \frac{1}{n}(x_k - x_{k-1}), \quad (70)$$

we can write

$$\begin{aligned}② &= \mathbb{E}[2\langle x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*, \beta r_k^i\rangle \mid x_k, \mathbf{S}_k] \\ &= 2\langle x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*, \beta\mathbb{E}[r_k^i \mid x_k, \mathbf{S}_k]\rangle \\ &\overset{(70)}{=} 2\langle x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) - x_*, \tfrac{\beta}{n}(x_k - x_{k-1})\rangle \\ &= 2\tfrac{\beta}{n}\langle x_k - x_*, x_k - x_{k-1}\rangle + 2\omega\tfrac{\beta}{n}\langle\nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle \\ &= 2\tfrac{\beta}{n}\langle x_k - x_*, x_k - x_*\rangle + 2\tfrac{\beta}{n}\langle x_k - x_*, x_* - x_{k-1}\rangle + 2\omega\tfrac{\beta}{n}\langle\nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle \\ &= 2\tfrac{\beta}{n}\|x_k - x_*\|^2 + 2\tfrac{\beta}{n}\langle x_k - x_*, x_* - x_{k-1}\rangle + 2\omega\tfrac{\beta}{n}\langle\nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle. \quad (71)\end{aligned}$$

Using the fact that for arbitrary vectors $a, b, c \in \mathbb{R}^n$ we have the identity $2\langle a-c, c-b\rangle = \|a-b\|^2 - \|c-b\|^2 - \|a-c\|^2$, we obtain

$$2\langle x_k - x_*, x_* - x_{k-1}\rangle = \|x_k - x_{k-1}\|^2 - \|x_{k-1} - x_*\|^2 - \|x_k - x_*\|^2.$$

Substituting this into (71) gives

$$\textcircled{2} = \tfrac{\beta}{n}\|x_k - x_*\|^2 + \tfrac{\beta}{n}\|x_k - x_{k-1}\|^2 - \tfrac{\beta}{n}\|x_{k-1} - x_*\|^2 + 2\omega\tfrac{\beta}{n}\langle \nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle. \tag{72}$$

The third expression can be bound as

$$\begin{aligned}
\textcircled{3} &= \mathbb{E}[\beta^2\|r_k^i\|^2 \mid x_k, \mathbf{S}_k] \\
&= \beta^2 \mathbb{E}_i[\|r_k^i\|^2] \\
&= \beta^2 \sum_{i=1}^n \tfrac{1}{n}(x_k^i - x_{k-1}^i)^2 \\
&= \tfrac{\beta^2}{n}\|x_k - x_{k-1}\|^2 \\
&= \tfrac{\beta^2}{n}\|(x_k - x_*) + (x_* - x_{k-1})\|^2 \\
&\le \tfrac{2\beta^2}{n}\|x_k - x_*\|^2 + \tfrac{2\beta^2}{n}\|x_{k-1} - x_*\|^2.
\end{aligned} \tag{73}$$

By substituting the bounds (69), (72), (73) into (68) we obtain

$$\begin{aligned}
\mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k, \mathbf{S}_k] &\le \|x_k - x_*\|^2 - 2\omega(2-\omega) f_{\mathbf{S}_k}(x_k) \\
&\quad + \tfrac{\beta}{n}\|x_k - x_*\|^2 + \tfrac{\beta}{n}\|x_k - x_{k-1}\|^2 - \tfrac{\beta}{n}\|x_{k-1} - x_*\|^2 \\
&\quad + 2\omega\tfrac{\beta}{n}\langle \nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle + 2\tfrac{\beta^2}{n}\|x_k - x_*\|^2 \qquad (74) \\
&\quad + 2\tfrac{\beta^2}{n}\|x_{k-1} - x_*\|^2 \\
&\overset{(54)}{\le} \left(1 + 3\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_k - x_*\|^2 + \left(\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_{k-1} - x_*\|^2 \\
&\quad - 2\omega(2-\omega) f_{\mathbf{S}_k}(x_k) + 2\omega\tfrac{\beta}{n}\langle \nabla f_{\mathbf{S}_k}(x_k), x_{k-1} - x_k\rangle. \qquad (75)
\end{aligned}$$

We now take the middle expectation (see (67)) and apply it to inequality (75):

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k, \mathbf{S}_k] \mid x_k] &\le \left(1 + 3\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_k - x_*\|^2 + \left(\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_{k-1} - x_*\|^2 \\
&\quad - 2\omega(2-\omega) f(x_k) + 2\omega\tfrac{\beta}{n}\langle \nabla f(x_k), x_{k-1} - x_k\rangle \\
&\le \left(1 + 3\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_k - x_*\|^2 + \left(\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_{k-1} - x_*\|^2 \\
&\quad - 2\omega(2-\omega) f(x_k) + 2\omega\tfrac{\beta}{n}(f(x_{k-1}) - f(x_k)) \\
&= \left(1 + 3\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_k - x_*\|^2 + \left(\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n}\right)\|x_{k-1} - x_*\|^2 \\
&\quad - \left(2\omega(2-\omega) + 2\omega\tfrac{\beta}{n}\right) f(x_k) + 2\omega\tfrac{\beta}{n} f(x_{k-1}).
\end{aligned}$$

where in the second step we used the inequality $\langle \nabla f(x_k), x_{k-1} - x_k\rangle \le f(x_{k-1}) - f(x_k)$ and the fact that $\omega\beta \ge 0$, which follows from the assumptions. We now apply inequalities (48) and (49), obtaining

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k, \mathbf{S}_k] \mid x_k] &\le \underbrace{\left(1 + 3\tfrac{\beta}{n} + 2\tfrac{\beta^2}{n} - \left(\omega(2-\omega) + \omega\tfrac{\beta}{n}\right)\lambda_{\min}^+\right)}_{a_1}\|x_k - x_*\|^2 \\
&\quad + \underbrace{\tfrac{1}{n}\left(\beta + 2\beta^2 + \omega\beta\lambda_{\max}\right)}_{a_2}\|x_{k-1} - x_*\|^2.
\end{aligned}$$

By taking expectation again (outermost expectation in the tower rule (67)), and letting $F_k := \mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2]$, we get the relation

$$F_{k+1} \leq a_1 F_k + a_2 F_{k-1}. \tag{76}$$

It suffices to apply Lemma 9 to the relation (55). The conditions of the lemma are satisfied. Indeed, $a_2 \geq 0$, and if $a_2 = 0$, then $\beta = 0$ and hence $a_1 = 1 - \omega(1-\omega)\lambda_{\min}^+ > 0$. The condition $a_1 + a_2 < 1$ holds by assumption.

The convergence result in function values follows as a corollary by applying inequality (48) to (30).

References

[1] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *ICML*, pages 1110–1119, 2016.

[2] D.P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

[3] D. Blatt, A.O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 14(SI):2508–2530, 2006.

[5] C.L. Byrne. *Applied iterative methods*. AK Peters Wellesley, 2008.

[6] A. Chambolle, M.J. Ehrhardt, P. Richtárik, and C.B. Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *arXiv preprint arXiv:1706.04957*, 2017.

[7] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[8] D. Csiba and P. Richtárik. Global convergence of arbitrary-block gradient methods for generalized polyak-lojasiewicz functions. *arXiv preprint arXiv:1709.03014*, 2017.

[9] Nair Maria Maia De Abreu. Old and new results on algebraic connectivity of graphs. *Linear Algebra and its Applications*, 423(1):53–73, 2007.

[10] C. De Sa, B. He, I. Mitliagkas, C. Ré, and P. Xu. Accelerated stochastic power iteration. *arXiv preprint arXiv:1707.02670*, 2017.

[11] A. Defazio. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems*, pages 676–684, 2016.

[12] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[13] A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[14] S. Elaydi. *An Introduction to Difference Equations*. Springer Science & Business Media, 2005.

[15] Y.C. Eldar and D. Needell. Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma. *Numerical Algorithms*, 58(2):163–177, 2011.

[16] O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[17] J.P. Fillmore and M.L. Marx. Linear recursive sequences. *SIAM Review*, 10(3):342–353, 1968.

[18] S. Gadat, F. Panloup, and S. Saadane. Stochastic heavy ball. *arXiv:1609.04228*, 2016.

[19] S. Geman. A limit theorem for the norm of random matrices. *The Annals of Probability*, pages 252–261, 1980.

[20] E. Ghadimi, H.R. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In *Control Conference (ECC), 2015 European*, pages 310–315. IEEE, 2015.

[21] E. Ghadimi, I. Shames, and M. Johansson. Multi-step gradient methods for networked optimization. *IEEE Transactions on Signal Processing*, 61(21):5417–5429, 2013.

[22] R.M. Gower, D. Goldfarb, and P. Richtárik. Stochastic block BFGS: squeezing more curvature out of data. In *ICML*, pages 1869–1878, 2016.

[23] R.M. Gower and P. Richtárik. Randomized iterative methods for linear systems. *SIAM. J. Matrix Anal. & Appl.*, 36(4):1660–1690, 2015.

[24] R.M. Gower and P. Richtárik. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890*, 2015.

[25] R.M. Gower and P. Richtárik. Linearly convergent randomized iterative methods for computing the pseudoinverse. *arXiv preprint arXiv:1612.06255*, 2016.

[26] R.M. Gower and P. Richtárik. Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. *arXiv preprint arXiv:1602.01768*, 2016.

[27] M. Gurbuzbalaban, A. Ozdaglar, and P.A. Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM J. Optim.*, 27(2):1035–1048, 2017.

[28] F. Hanzely, J. Konečný, N. Loizou, P. Richtárik, and D. Grishchenko. Privacy preserving randomized gossip algorithms. *arXiv preprint arXiv:1706.07636*, 2017.

[29] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[30] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de lAcademie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.

[31] J. Konečný, J. Liu, P. Richtárik, and M. Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2016.

[32] J. Konečný and P. Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3(9):1–14, 2017.

[33] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[34] Y.T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 147–156. IEEE, 2013.

[35] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim.*, 26(1):57–95, 2016.

[36] D. Leventhal and A.S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[37] J. Liu and S. Wright. An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation*, 85(297):153–178, 2016.

[38] N. Loizou and P. Richtárik. A new perspective on randomized gossip algorithms. In *4th IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.

[39] Nicolas Loizou and Peter Richtárik. Linearly convergent stochastic heavy ball method for minimizing generalization error. *arXiv preprint arXiv:1710.10737*, 2017.

[40] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1590–1604, 2015.

[41] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50(2):395–403, 2010.

[42] D. Needell, N. Srebro, and R. Ward. Stochastic gradient descent and the randomized Kaczmarz algorithm. *Mathematical Programming, Series A*, 155(1):549–573, 2016.

[43] D. Needell and J.A. Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra and its Applications*, 441:199–221, 2014.

[44] D. Needell, R. Zhao, and A. Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra and its Applications*, 484:322–343, 2015.

[45] A. Nemirovskii and D.B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley Interscience, 1983.

[46] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[47] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[48] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

[49] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641, 2015.

[50] J. Nutini, B. Sepehry, I. Laradji, M. Schmidt, H. Koepke, and A. Virani. Convergence rates for greedy Kaczmarz algorithms, and faster randomized Kaczmarz rules using the orthogonality graph. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 547–556. AUAI Press, 2016.

[51] P. Ochs, T. Brox, and T. Pock. ipiasco: Inertial proximal algorithm for strongly convex optimization. *Journal of Mathematical Imaging and Vision*, 53(2):171–181, 2015.

[52] P. Ochs, Y. Chen, T. Brox, and T. Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[53] M. Penrose. *Random Geometric Graphs*. Number 5. Oxford University Press, 2003.

[54] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[55] B.T. Polyak. Introduction to optimization. translations series in mathematics and engineering. *Optimization Software*, 1987.

[56] C. Popa. Least-squares solution of overdetermined inconsistent linear systems using Kaczmarz's relaxation. *International Journal of Computer Mathematics*, 55(1-2):79–89, 1995.

[57] C. Popa. Convergence rates for Kaczmarz-type algorithms. *arXiv preprint arXiv:1701.08002*, 2017.

[58] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

[59] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling ii: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

[60] Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq. SDNA: Stochastic dual Newton ascent for empirical risk minimization. *ICML*, 2016.

[61] Z. Qu, P. Richtárik, and T. Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in neural information processing systems*, pages 865–873, 2015.

[62] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[63] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

[64] P. Richtárik and M. Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv:1706.01108*, 2017.

[65] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[66] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Math. Program.*, 162(1-2):83–112, 2017.

[67] F. Schöpfer and D.A. Lorenz. Linear convergence of the randomized sparse Kaczmarz method. *arXiv preprint arXiv:1610.02889*, 2016.

[68] Sh. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. *JMLR*, 14(1):567–599, 2013.

[69] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262–278, 2009.

[70] I. Sutskever, J. Martens, G.E. Dahl, and G.E. Hinton. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28:1139–1147, 2013.

[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[72] P. Tseng. An incremental gradient (-projection) method with momentum term and adaptive stepsize rule. *SIAM J. Optim.*, 8(2):506–531, 1998.

[73] S. Tu, S. Venkataraman, A.C. Wilson, A. Gittens, M.I. Jordan, and B. Recht. Breaking locality accelerates block Gauss-Seidel. In *ICML*, 2017.

[74] A.C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv:1705.08292*, 2017.

[75] S.J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[76] H. Xiang and L. Zhang. Randomized iterative methods with alternating projections. *arXiv preprint arXiv:1708.09845*, 2017.

[77] T. Yang, Q. Lin, and Z. Li. Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization. *arXiv preprint arXiv:1604.03257*, 2016.

[78] S.K. Zavriev and F.V. Kostyuk. Heavy-ball method in nonconvex optimization problems. *Computational Mathematics and Modeling*, 4(4):336–341, 1993.

[79] J. Zhang, I. Mitliagkas, and C. Ré. Yellowfin and the art of momentum tuning. *arXiv preprint arXiv:1706.03471*, 2017.

[80] A. Zouzias and N.M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM. J. Matrix Anal. & Appl.*, 34(2):773–793, 2013.